全国高等职业教育计算机类规划教材·工作过程系统化教程系列

软件测试项目实战

于艳华　王素华　主　编

吴艳平　岳淑玲　副主编

陈显刚　李　季　主　审

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

VISIT…

LANZAROTE Caliente.com

内 容 简 介

本书是一本体现工作过程系统化的实战书。全书以软件测试企业的真实测试流程为主线，按软件测试企业流程详细地介绍了测试计划、测试用例、测试执行、测试总结、测试工具等五部分内容。

本教材聘请了在企业中从事软件测试工作多年的专家孙会生先生为企业顾问，通过企业调研，确定了本书的流程与内容，真正体现了校企合作与工作过程系统的理念。本书以网上购物系统为参照，讲解了软件测试的流程，真实体现了一个项目从开始到结束测试的整个过程。同时以天天超市管理系统为拓展进行了内容的巩固。让学生在模拟工作环境中学会处理各类问题的方法，实现理论与实践一体化教学，也真正把培养学生的方法能力放在了首位。

本书同时提供了教材中所用项目的测试用例及配套电子课件。本书可作为高职高专计算机专业及相关非计算机专业的教材使用，也可作为培训教材及对软件测试感兴趣的初学者入门使用。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

软件测试项目实战 / 于艳华，王素华主编．—北京：电子工业出版社，2009.7
全国高等职业教育计算机类规划教材 • 工作过程系统化教程系列
ISBN 978-7-121-08967-1

Ⅰ. 软…　Ⅱ. ①于…②王…　Ⅲ. 软件－测试－高等学校：技术学校－教材　Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2009）第 086180 号

策划编辑：程超群
责任编辑：徐云鹏
印　　刷：
装　　订：
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×1 092　1/16　印张：14　字数：358.4 千字
印　　次：2009 年 7 月第 1 次印刷
印　　数：4 000 册　　定价：22.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。
服务热线：（010）88258888。

前　　言

作为全国高职示范院校之一，我院紧跟高职课程改革步伐，密切与企业合作，按照工作过程系统化的理念，对课程进行了一系列的改革与实践，取得了初步的成果。软件测试课程是我们改革的课程之一，通过与企业专家座谈，与企业合作，聘请企业专家参与教材的编写等一系列措施，共同完成了本教材的编写工作。

以“工学结合”为切入点，精选企业真实项目，提炼出以职业能力培养为特色的教学内容，突出技术实用性。充分考虑高职学生认知能力，对每个知识点的讲解采用“任务驱动+自学辅导教学”的方法，本书从企业实际工作过程的角度出发，全面系统地讲解了在企业中，针对一个软件项目进行测试活动的一个完整的测试流程，通过本书的讲解，使初学者能够很快地理解什么是软件测试，如何编写测试计划、测试用例、测试执行、测试总结一系列测试活动。是手把手教你学习软件测试的一个很好的工具。

本书的特色：聘请了企业测试专家参与编写，以一个项目完整的测试工作过程来设计本书的流程，完全按照企业的流程去设计，除了一个教学项目外，还有一个拓展项目让你来巩固所学的内容。

本书共分为五章。以一个项目为教学项目，另一个教学项目为拓展项目，讲解了软件测试的企业流程、方法、技术等内容。

测试计划　是一个测试活动成功与否的关键，本章以一个具体项目为例讲解了软件测试计划的设计，及软件测试计划所包括的要素。

测试用例　根据测试项目网上购物系统来设计测试用例，测试用例设计完整，充分，为测试执行做好了充分的准备。

测试执行　根据测试计划及测试用例，来设计测试执行，这里用的是 LoadRunner 8.0 测试工具来进行的自动化测试执行过程。

测试总结　由第 3 章测试执行来分析测试结果。

测试工具　详细介绍了 LoadRunner 8.0 的安装，参数设置、场景设计等内容。

本书由长春职业技术学院于艳华、王素华任主编，吴艳平、岳淑玲任副主编，刘宝庆、孙会生（吉大正元信息技术股份有限公司）、丛路卫、刘孝国（吉林农业大学）参与了本书的编写工作。陈显刚、李季担任本教材主审工作。

由于作者水平和时间有限，书中难免有错误之处，欢迎各界同事给予批评指正 E-mail：spbyyh@163.com.

编　者

2009 年 4 月

目　　录

第 1 章　测试计划 …… (1)
工作任务 1.1　项目任务说明 …… (1)
1.1.1　项目任务工作流程总体说明 …… (1)
1.1.2　应知应会 …… (2)
1.1.3　拓展任务 …… (6)
工作任务 1.2　测试计划 …… (6)
1.2.1　网上购物系统的测试计划 …… (6)
1.2.2　应知应会 …… (9)
1.2.3　拓展任务 …… (23)
第 2 章　测试用例 …… (24)
工作任务 2.1　Test Suite 用户管理 …… (24)
2.1.1　Test Suite 添加注册信息 …… (24)
2.1.2　Test Suite 管理员登录 …… (33)
2.1.3　Test Suite 注册用户登录 …… (36)
2.1.4　Test Suite 修改注册信息 …… (41)
2.1.5　应知应会 …… (47)
2.1.6　拓展任务 …… (60)
工作任务 2.2　Test Suite 商品管理 …… (60)
2.2.1　Test Suite 商品类别管理 …… (60)
2.2.2　Test Suite 商品管理 …… (65)
2.2.3　应知应会 …… (71)
2.2.4　拓展任务 …… (75)
工作任务 2.3　Test Suite 购物管理 …… (75)
2.3.1　Test Suite 商品查看 …… (75)
2.3.2　Test Suite 购买商品 …… (77)
2.3.3　应知应会 …… (80)
2.3.4　拓展任务 …… (84)
工作任务 2.4　Test Suite 订单管理 …… (84)
2.4.1　Test Suite 订单查询 …… (84)
2.4.2　Test Suite 订单查看 …… (85)
2.4.3　Test Suite 订单详情 …… (86)
2.4.4　应知应会 …… (87)
2.4.5　拓展任务 …… (91)
工作任务 2.5　Test Suite 其他测试 …… (91)
2.5.1　Test Suite 性能测试 …… (91)
2.5.2　Test Suite 链接测试 …… (94)

2.5.3　Test Suite 导航测试 …… （95）
2.5.4　Test Suite 界面测试 …… （96）
2.5.5　Test Suite 兼容性测试 …… （99）
2.5.6　Test Suite 帮助文档测试 …… （100）
2.5.7　应知应会 …… （102）
2.5.8　拓展任务 …… （116）
第 3 章　测试执行 …… （117）
工作任务 3.1　测试执行概述 …… （117）
3.1.1　测试执行概述 …… （117）
3.1.2　应知应会 …… （119）
3.1.3　拓展任务 …… （119）
工作任务 3.2　测试执行准备工作 …… （119）
3.2.1　对执行测试人员的培训 …… （119）
3.2.2　测试任务及进度的安排 …… （120）
3.2.3　自动化测试的执行 …… （120）
3.2.4　应知应会 …… （134）
3.2.5　拓展任务 …… （138）
工作任务 3.3　测试执行结果与分析 …… （138）
3.3.1　网上订票系统的测试结果与分析 …… （138）
3.3.2　应知应会 …… （141）
3.3.3　拓展任务 …… （141）
第 4 章　测试总结 …… （142）
工作任务 4.1 测试总结 …… （142）
4.1.1　网上购物系统的测试总结 …… （142）
4.1.2　应知应会 …… （153）
4.1.3　拓展任务 …… （162）
第 5 章　测试工具 …… （163）
工作任务 5.1　LoadRunner 8.0 的安装过程 …… （163）
5.1.1　LoadRunner 8.0 的安装过程 …… （163）
5.1.2　应知应会 …… （167）
5.1.3　拓展任务 …… （169）
工作任务 5.2　LoadRunner 8.0 生成测试脚本 …… （169）
5.2.1　LoadRunner 录制脚本 …… （169）
5.2.2　调试并完善脚本 …… （173）
5.2.3　应知应会 …… （181）
5.2.4　拓展任务 …… （181）
工作任务 5.3　软件测试场景 …… （182）
5.3.1　创建运行场景 …… （182）
5.3.2　应知应会 …… （190）
5.3.3　拓展任务 …… （192）

工作任务 5.4　LoadRunner 的结果分析……（192）
5.4.1　LoadRunner 调用 Analysis……（192）
5.4.2　应知应会……（194）
5.4.3　拓展任务……（195）
附录 A　常用测试工具清单……（196）
附录 B　基于测试概念进行代码设计的基本原则……（198）
附录 C　软件测试术语表……（200）
附录 D　参考文献……（215）

第 1 章　测试计划

本章介绍网上购物系统的测试计划书的编写。

本章重点：

- “5W1H”规则，明确内容与过程
- 网上购物系统的测试计划设计

撰写软件测试计划是软件测试流程中的第一个环节。软件测试的成功与失败是相对的，即要参照测试计划来判断。如果达到测试中预定的管理目标，则可以说软件测试是成功的，否则是失败的。软件测试计划一般是由测试经理来写的，但也不绝对，在小公司里面没有专门的测试部，一个项目组里面只有一个测试人员，就得由测试人员来写。

工作任务 1.1　项目任务说明

1.1.1　项目任务工作流程总体说明

1．工作任务描述

本书以工作过程系统化为设计理念，企业人员参与本书的设计，按照企业真实的测试流程设计各章节内容，将真实项目网上购物系统的测试活动贯穿始终，并辅以拓展项目天天超市管理系统，使学生能够更好地掌握测试流程，可以达到企业测试岗位技能的要求。

2．工作过程

（1）确定测试计划

根据用户需求报告中关于功能要求和性能指标的规格说明书，定义相应的测试需求报告，即制定黑盒测试的最高标准，以后所有的测试工作都将围绕着测试需求来进行，符合测试需求的应用程序即是合格的，反之即是不合格的；同时，还要适当选择测试内容，合理安排测试人员、测试时间及测试资源等。测试计划描述所要完成的测试，是指导测试的纲领性文件，根据不同公司对项目的不同要求，测试计划的内容不尽相同，但是主要内容大同小异。

本书将详细介绍网上购物系统的测试活动，第 1 章将详细介绍测试计划的编写。

（2）设计测试用例

测试用例（Test Case，缩写 TC），指的是在测试执行之前设计的一套详细的测试方案，包括测试环境、测试步骤、测试数据和预期结果。即：

测试用例=输入+输出+测试环境

其中，“输入”包括测试数据和测试步骤，“输出”指的是期望结果，而“测试环境”指的就是系统环境设置。

本书第 2 章将详细介绍网上购物系统各种用例的编写方法和实例。

（3）测试执行

测试执行是测试计划贯彻实施的保证，是测试用例实现的必然过程，严格地测试执行使测试工作不会半途而废。测试执行前，应做好如下准备工作：

① 测试环境的搭建

测试用例执行过程中，搭建测试环境是第一步。一般来说，软件产品提交测试后，开发人员应该提交一份产品安装指导书，在指导书中详细指明软件产品运行的软 硬件环境，比如要求操作系统是Windows 2000 版本，数据库是SQL Server 2000 等等，此外，应该给出被测试软件产品的详细安装指导书，包括安装的操作步骤、相关配置文件的配置方法等。

② 测试任务的安排

不仅包括指定哪些人参加测试活动，谁负责功能测试、性能测试、界面测试等，谁负责测试环境的维护等，还要包括人员的培训，知识的传递等。

③ 测试用例执行

测试执行过程中，当测试的实际输出结果与测试用例中的预期输出结果一致的时候，是否可以认为测试用例执行成功了？答案是否定的，即便实际测试结果与测试的预期结果一致，也要查看软件产品的操作日志、系统运行日志和系统资源使用情况，来判断测试用例是否执行成功了。全方位观察软件产品的输出可以发现很多隐蔽的问题。测试执行过程中，应该注意及时更新测试用例。往往在测试执行过程中，才发现遗漏了一些测试用例，这时候应该及时的补充；往往也会发现有些测试用例在具体的执行过程中根本无法操作，这时候应该删除这部分用例；也会发现若干个冗余的测试用例完全可以由某一个测试用例替代，那么删除冗余的测试用例。

④ 缺陷报告

缺陷报告单中最关键的几个部分：第一部分是发现缺陷的环境，包括软件环境、硬件环境等；第二部分是缺陷的基本描述；第三部分是开发人员对缺陷的解决方法。通过对上述缺陷报告单的三个部分进行仔细分析，从中掌握了软件产品最常见的基本问题，并吸收了其他软件测试人员的工作经验。

（4）测试总结

软件测试执行结束后，测试活动还没有结束。测试结果分析是必不可少的重要环节，“编筐编篓，全在收口”，测试结果的分析对下一轮测试工作的开展有很大的借鉴意义。前面的“测试准备工作”中，建议测试人员阅读缺陷跟踪库，查阅其他测试人员发现的软件缺陷。测试结束后，也应该分析自己发现的软件缺陷，对发现的缺陷分类，你会发现自己提交的问题只有固定的几个类别；然后，再把一起完成测试执行工作的其他测试人员发现的问题也汇总起来，你会发现，你所提交问题的类别与他们有差异。这很正常，人的思维是有局限性，在测试的过程中，每个测试人员都有自己思考问题的盲区和测试执行的盲区，有效的自我分析和分析其他测试人员，你会发现自己的盲区，有针对性的分析盲区，必定会在下一轮测试中避免盲区。

1.1.2 应知应会

1. 关于软件测试

（1）软件测试概述

软件测试是什么？软件测试就是对项目开发过程的产品（编码、文档等）进行差错审查，

保证其质量的一种过程。

软件业的迅猛发展也就是近几十年的过程，时间虽短，但许多误解似乎已根深蒂固，对测试的偏见也是如此。“软件的重点在于需求、在于分析、在于设计、在于开发，而测试，容易，没什么技术含量，找一些用户，对照需求尽力去测就行了；有时间多测点，没时间就少测点。”这种看法在许多项目经理、软件负责人的心中固守着，难以改变。

这种观念的结果有目共睹，是什么？很简单，是大量软件 Bug、缺陷的“流失”，从测试人员手中悄然而过，流失到用户手中，流失到项目维护阶段。随之而来的，便是用户无休止的抱怨、维护人员无休止的“救火”、维护成本无休止的增加。这是软件人员的梦魇！恶梦总有醒来时，经过无数教训的重击，在不堪回首而不得不回首的经历中，软件业的管理者发现：是他们错了，软件测试是不可忽视的。

“所有这些问题，假如在项目中测试到的话，便不会有造成不可收拾的结果了。”——人们终于意识到测试简单而纯真的真谛。

软件测试从直观上来讲是对测试对象进行检查、验证，似乎很简单，但实际不然，它是由许多处理环节构成的。根据测试目标、质量控制的要求，它被划分为以下各类环节，并被设置了不同的准入、准出标准。

（2）软件测试原则

① 尽早和持续不断的测试

② 彻底完全的测试是不可能的

③ 软件测试是有风险的行为

④ 并非所有的软件错误都能修复

⑤ 反向思维逻辑

⑥ 由小到大的测试范围

⑦ 避免测试自己的项目

⑧ 从用户需求入手

（3）为什么不能完全测试

① 测试数据输入量太大

② 输出结果太多

③ 软件的操作步骤太多

④ 软件说明书并非“盲人手册”

（4）并非所有的错误都能修复，Bug 不能被关闭的原因

① 不算真正的软件错误

② 没有足够的时间

③ 修复的风险太大

④ 不值得修复

（5）错误集中发生现象

① 软件开发人员的疲劳，造成大量代码坏块

② 程序人员往往会犯同样的错误，因为大部分代码都是复制、粘贴而来

③ 软件的基础构架问题，有些软件的底层支撑系统因为“年久失修”变得越来越力不从心了

④ 发现缺陷的时间越早，Bug 所造成的损失会越小

（6）避免检查自己的代码的原因

① 程序员从来都不会承认自己写的程序有错误

② 程序员的测试思路有明显的局限性

③ 多数程序员没有经过严格正规的职业训练

④ 程序员无良好的 Bug 跟踪和回归测试经验

2．测试过程

测试过程应该建立的测试案例，并按严格的流程来进行，图 1-1 就是一个测试案例的使用流程，是严格意义上的测试过程。

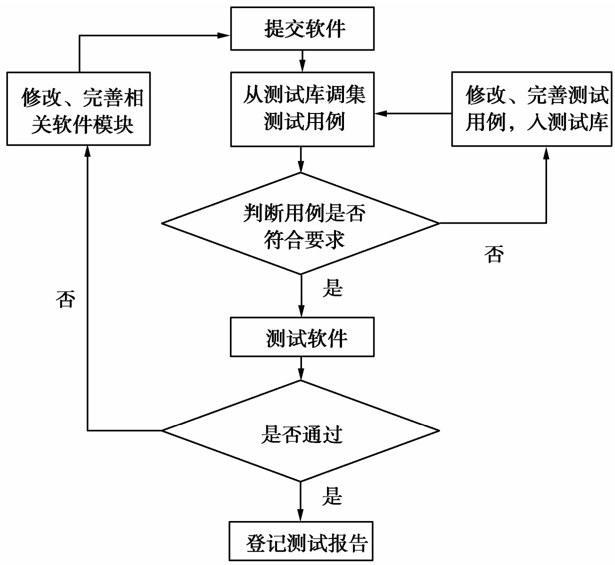


图 1-1　测试流程

正常的测试案例使用方式如图 1-1 所示，测试设计阶段，相关测试设计人员会对测试对象进行了解、分析，为保证测试顺利进行，保证测试覆盖尽量多的测试对象，会设计测试案例、测试方案，在测试期间进行使用；测试发现错误时，软件技术人员会根据测试的缺陷反馈结果及技术人员的软件修改信息对测试程序进行修改，完毕后再进行回归测试。

但是由于软件测试这个行业本身就兴起较晚，现在仍然处于比较不规范且存在很多问题尚未解决的阶段。传统的测试过程，测试管理不严密，测试人员未建立完整的测试库，未将测试案例、测试程序、测试方案进行有效保存，等到回归测试时，相关测试程序等往往已不知去向，无处可寻了；即使能找到这些程序、案例，可往往因为回归测试过于频繁、项目期限日益迫近，已经没有时间来修改、完善这些程序及案例，只能凭借经验、记忆及技术人员的口述对程序修改过的地方草草重测一遍而已，缺乏正规化的测试过程，造成测试的虎头蛇尾。

3．关于测试计划

软件测试计划是指导测试过程的纲领性文件，包含了产品概述、测试策略、测试方法、测试区域、测试配置、测试周期、测试资源、测试交流、风险分析等内容。借助软件测试计划，参与测试的项目成员，尤其是测试管理人员，可以明确测试任务和测试方法，保持测试实施过程的顺畅沟通，跟踪和控制测试进度，应对测试过程中的各种变更。

测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。所以其中最重要的是测试策略和测试方法（最好是能先评审）。

做好测试计划工作的关键是什么？

（1）明确测试的目标，增强测试计划的实用性

编写软件测试计划的重要目的就是使测试过程能够发现更多的软件缺陷，因此软件测试计划的价值取决于它对帮助管理测试项目，并且找出软件潜在的缺陷。因此，软件测试计划中的测试范围必须高度覆盖功能需求，测试方法必须切实可行，测试工具具有较高的实用性，便于使用，生成的测试结果直观、准确。

（2）坚持“5W”规则，明确内容与过程

（3）采用评审和更新机制，保证测试计划满足实际需求

测试计划设计完成后，如果没有经过评审，直接发送给测试团队，测试计划可能内容不准确或遗漏测试内容，或者软件需求变更引起测试范围的增减，而测试计划的内容没有及时更新，误导测试执行人员。

（4）分别创建测试计划与测试详细规格、测试用例

应把详细的测试技术指标包含到独立创建的测试详细规格文档中，把用于指导测试小组执行测试过程的测试用例放到独立创建的测试用例文档或测试用例管理数据库中。

4．测试用例的基本格式

软件测试用例的基本要素包括测试用例编号、测试标题、重要级别、测试输入、操作步骤、预期结果，下面逐一介绍。

用例编号：测试用例的编号有一定的规则，比如系统测试用例的编号定义规则为：WEBLOAD1-ST-001，命名规则是项目名称+测试阶段类型（系统测试阶段）+编号。定义测试用例编号，便于查找测试用例，便于测试用例的跟踪。

测试标题：对测试用例的描述，测试用例标题应该清楚表达测试用例的用途。比如“测试用户登录时输入了错误密码，软件的响应情况”。

重要级别：定义测试用例的优先级别，可以笼统地分为“高”和“低”两个级别。一般来说，如果软件需求的优先级为“高”，那么针对该需求的测试用例优先级也为“高”；反之亦然。

测试输入：提供测试执行中的各种输入条件。根据需求中的输入条件，确定测试用例的输入。测试用例的输入对软件需求当中的输入有很大的依赖性，如果软件需求中没有很好的定义需求的输入，那么测试用例设计中会遇到很大的障碍。

操作步骤：提供测试执行过程的步骤。对于复杂的测试用例，测试用例的输入需要分为几个步骤完成。

预期结果：提供测试执行的预期结果，预期结果应该根据软件需求中的输出得出。如果在实际测试过程中，得到的实际测试结果与预期结果不符，那么测试不通过；反之则测试通过。

软件测试用例的设计主要从上述 6 个方面考虑，结合相应的软件需求文档，在掌握一定测试用例设计方法的基础上，可以设计出比较全面、合理的测试用例。具体的测试用例设计方法可以参见相关的测试书籍，白盒测试方法和黑盒测试方法在绝大多数的软件测试书籍中都有详细的介绍，这里不做赘述。

1.1.3　拓展任务

（1）天天超市管理系统项目说明

天天超市管理系统主要包括两大模块，即采购模块、销售模块

（2）天天超市管理系统项目测试流程

工作任务 1.2　测试计划

1.2.1　网上购物系统的测试计划

1．工作任务描述

掌握测试计划的编写规范，测试计划的内容、格式、规则，初步设计测试计划。

2．工作过程

网上购物系统的测试计划（如表 1-1 所示）。

表 1-1　网上购物系统测试计划

日　　期	版　　本	说　　明	作　　者
2008-12-24	第一版	产品发布前，依据产品需求说明书制定本计划	于艳华

目　　录

一、概述
1.1　测试目的
1.2　测试范围
1.3　限制条件
1.4　参考文档
二、测试摘要
2.1　测试目标
2.2　资源和工具
2.2.1　资源
2.2.2　工具
2.3　送测要求
2.4　测试种类
三、测试风险
四、暂停标准和再启动要求
五、测试任务和进度
六、测试提交物

一、概述

1.1　测试目的

为了真实地模拟企业测试过程，我们将以“网上购物系统”为测试对象，展开系统测试。在测试前期，依据产品需求说明书设计测试用例。在产品开发结束后，适当地调整测试计划和测试用例，带领同学们执行测试用例，完成系统测试任务。

1.2　测试范围

本测试计划是针对《网上购物系统》.doc 和《程序测试规范》.doc 中规定的内容来制定的，包括：

- 用户管理
- 商品管理
- 购物管理
- 订单管理

1.3　限制条件

本次测试计划受限于产品开发人员提交测试的内容和提交时间。根据开发人员提交模块的实际情况，本计划会做出相应修改。

1.4　参考文档（如表 1-2 所示）

表 1-2　参考文档

序　　号	名　　称	作　　者	备　　注
1	《程序测试规范》.doc		
2	《网上购物系统》.doc		

二、测试摘要

2.1 测试目标

通过测试，达到以下目标：

- 测试已实现的产品是否达到设计的要求，包括：各个功能点是否业已实现，业务流程是否正确。
- 产品是否运行稳定，系统性能是否在可接受范围。
- Bug 数和缺陷率是否控制在可接受的范围之内，产品能否发布。

2.2 资源和工具

2.2.1 资源

- 测试服务器硬件配置：
  - 软件配置：
  - IP 地址：
- 人员：测试审核人 3 名，测试实施人员 30 名。

2.2.2 工具

- 缺陷管理工具：Mantis
- 链接检测工具：Xenu
- 自动化性能测试工具：LoadRunner

2.3 送测要求

提交的测试产品按以下要求进行（如表 1-3 所示）

表 1-3 测试产品要求说明

步　骤	动　作	负 责 人	相关文档或记录	要　求
1	打包、编译	开发人员	无	确认可测试
2	审核并提交测试	产品经理	审核报告	产品经理审核并签字
3	接收测试	测试负责人	接收任务单	确认产品有无重大缺陷，是否可以继续测试
4	执行测试	测试负责人	Bug 记录、测试总结报告	对产品质量做出评价

2.4 测试种类

计划完成以下类型测试

- 功能测试
- 界面测试
- 链接测试
- 兼容性测试
- 性能测试

三、测试风险

本次测试过程，受以下条件制约：

- Bug 的修复情况
- 模块功能的实现情况
- 系统整体功能的实现情况
- 代码编写的质量
- 人员经验以及对软件的熟悉度

➢ 人员调整导致研发周期延迟
➢ 测试时间的缩短导致某些测试计划无法执行

四、暂停标准和再启动要求

➢ 冒烟测试，发现一级错误（大于等于 1）、二级错误（大于等于 2）暂停测试返回开发。
➢ 软件项目需暂停以进行调整时，测试应随之暂停，并备份暂停点数据。
➢ 软件项目在其开发生命周期内出现重大估算，进度偏差，需暂停或终止时，测试应随之暂停或终止，并备份暂停或终止点数据。
➢ 如有新的项目需求，则在原测试计划下做相应的调整。
➢ 若开发暂停，则相应测试也暂停，并备份暂停点数据。
➢ 若项目中止，则对已完成的测试工作做测试活动总结。
➢ 项目再启动时，测试进度重新安排或顺延。

五、测试任务和进度（如表 1-4 所示）

表 1-4　测试任务及人员分配表

测 试 阶 段	测 试 任 务	工作量估计	人 员 分 配	起 止 时 间
第一阶段	功能测试	2 日	每名教师带领 10 人	
第二阶段	界面测试	1 日	1 名教师带领全部学生	
第三阶段	链接测试	1 日	1 名教师带领全部学生	
第四阶段	兼容性测试	1 日	每名教师带领 10 人	
第五阶段	性能测试	2 日	每名教师带领 10 人	
第六阶段	测试总结	1 日	测试负责人	

六、测试提交物

本次测试需要提交：

➢ 测试计划
➢ 测试用例
➢ 缺陷记录
➢ 测试总结

编 制 人：于 艳 华
编制日期：2008-12-24

1.2.2　应知应会

1．关于测试计划

俗话说：凡事预则立，不预则废！软件测试同样，在测试 项目之初就要制定相应的测试计划。接下来了解一下如何 编写测试计划。

（1）为什么要编写测试计划

① 领导能够根据测试计划做宏观调控，进行相应资源配置等；
② 测试人员能够了解整个项目测试情况以及项目测试不同阶段所要进行的工作等；
③ 便于其他人员了解测试人员的工作内容，进行有关配合工作。

（2）什么时间开始编写测试计划

软件测试计划在项目启动初期就应该规划。

（3）由谁来编写测试计划

软件测试计划一般由具有丰富经验的项目测试负责人来编写。

（4）测试计划编写 6 要素（5W1H）

① why——为什么要进行这些测试；

② what——测试哪些方面，不同阶段的工作内容；

③ when——测试不同阶段的起止时间；

④ where——相应文档，缺陷的存放位置，测试环境等；

⑤ who——项目有关人员组成，安排哪些测试人员进行测试；

⑥ how——如何去做，使用哪些测试 工具以及测试方法进行测试。

2．测试计划模板

因为各个公司的测试计划模板是不同的，这是一个比较完整的测试计划模板，写得很详细（见表 1-5、表 1-6），学生可以参考完成天天超市管理系统的测试计划。

项目名称（项目编号）

测试计划

（部门名称）

×××软件公司

表 1-5　测试计划说明表

总页数		正文		附录		生效日期：　年　月　日
编制：			审核：			批准：

表 1-6　修订历史记录

日　期	版　本	说　明	作　者
<日/月/年>	<x.x>	<详细信息>	<姓名>

目　　录

一、简介
二、测试需求
三、测试风险
四、测试策略
五、工具
六、资源
七、测试进度和里程碑
八、可交付工件

一、简介

1．目的

<项目名称>的“测试计划”文档的目的是：

（1）提供一个对项目软件进行测试的总体安排和进度计划，确定现有项目的信息和应测试软件构件。

（2）标明推荐的测试需求（高层次）。

（3）推荐可采用的测试策略，并对这些策略加以说明。

（4）确定所需的资源，并对测试的工作量进行估计。

（5）列出测试项目的可交付元素。

2．背景

输入测试对象（组件、应用程序、系统等）及其目标的简要说明。需要包括的信息有：主要的功能和特性、测试对象的构架以及项目的简史。本部分应该只包含 3 至 5 个段落。

3．范围

描述测试的各个阶段，例如：单元测试、集成测试或系统测试，并说明本计划所针对的测试类型（如功能测试或性能测试）。简要地列出测试对象中将接受测试或将不接受测试的那些特性和功能。

如果在编写此文档的过程中作出的某些假设可能会影响测试设计、开发或实施，则列出所有这些假设。

列出可能会影响测试设计、开发或实施的所有风险或意外事件。

列出可能会影响测试设计、开发或实施的所有约束。

4．使用文档

表 1-7 列出了制定测试计划所用的文档，并标明了文档的可用性：

注：可以视情况删除或添加项目。

表 1-7　测试计划使用文档列表

文档（版本/日期）	已创建或可用	已被接受或已经过复审	作者或来源	备　注
需求规约	□ 是　□ 否	□ 是　□ 否		
功能性规约	□ 是　□ 否	□ 是　□ 否		
用例报告	□ 是　□ 否	□ 是　□ 否		
项目计划	□ 是　□ 否	□ 是　□ 否		
设计规约	□ 是　□ 否	□ 是　□ 否		

续表

文档（版本/日期）	已创建或可用	已被接受或已经过复审	作者或来源	备　注
原型	□ 是 □ 否	□ 是 □ 否		
用户手册	□ 是 □ 否	□ 是 □ 否		
业务模型或业务流程	□ 是 □ 否	□ 是 □ 否		
数据模型或数据流	□ 是 □ 否	□ 是 □ 否		
业务功能和业务规则	□ 是 □ 否	□ 是 □ 否		
项目或业务风险评估	□ 是 □ 否	□ 是 □ 否		

二、测试需求

下面列出了那些已被确定为测试对象的项目（用例、功能性需求和非功能性需求）。

（1）数据库测试；

（2）功能性测试；

（3）业务周期测试；

（4）用户界面测试；

（5）性能测试；

（6）负载测试；

（7）强度测试；

（8）容量测试；

（9）安全性和访问控制测试；

（10）故障转移/恢复测试；

（11）配置测试；

（12）安装测试。

三、测试风险

软件测试 风险是不可避免的、总是存在的，所以对测试风险的管理非常重要，必须尽力降低测试中所存在的风险，最大程度地保证质量和满足客户的需求。在测试工作中，主要的风险有：

（1）质量需求或产品的特性理解不准确，造成测试范围分析的误差，结果某些地方始终测试不到或验证的 标准不对。

（2）测试用例没有得到百分之百的执行，如有些测试用例被有意或无意的遗漏。

（3）需求的临时或突然变化，导致设计的修改和代码的重写，测试时间不够。

（4）质量标准不都是很清晰的，如适用性的测试，仁者见仁、智者见智。

（5）测试用例设计不到位，忽视了一些边界条件、深层次的逻辑、用户场景等。

（6）测试环境，一般不可能和实际运行环境完全一致，造成测试结果的误差。

（7）有些缺陷出现频率不是百分之百，不容易被发现；如果代码质量差，软件缺陷很多，被漏检的缺陷可能性就大。

（8）回归测试一般不运行全部测试用例，是有选择性的执行，必然带来风险。

前面三种风险是可以避免的，而 4 至 7 的四种风险是不能避免的，可以降到最低。最后一种回归测试风险是可以避免，但出于时间或成本的考虑，一般也是存在的。

针对上述软件测试的风险，有一些有效的测试风险控制方法，如：

测试环境不对可以通过事先列出要检查的所有条目，在测试环境设置好后，由其他人员按已列出条目逐条检查。

有些测试风险可能带来的后果非常严重，能否将它转化为其他一些不会引起严重后果的低风险。如产品发布前夕，在某个不是很重要的新功能上发现一个严重的缺陷，如果修正这个缺陷，很有可能引起某个原有功能上的缺陷。这时处理这个缺陷所带来的风险就很大，对策是去掉那个新功能，转移这种风险。

有些风险不可避免，就设法降低风险，如“程序中未发现的缺陷”这种风险总是存在，我们就要通过提高测试用例的覆盖率（如达到 99.9%）来降低这种风险。

为了避免、转移或降低风险，事先要做好风险管理计划和控制风险的策略，并对风险的处理还要制定一些应急的、有效的处理方案。

四、测试策略

测试策略提供了推荐用于测试对象的方法。第二部分“测试需求”中说明了将要测试哪些对象，而本部分则要说明如何对测试对象进行测试。对于每种测试，都应提供测试说明，并解释其实施和执行的原因。如果不实施和执行某种测试，则应该用一句话加以说明，并陈述这样做的理由。例如，“将不实施和执行该测试。该测试不合适。”制定测试策略时所考虑的主要事项有：将要使用的方法以及判断测试何时完成的标准。下面列出了在进行每项测试时需考虑的事项，除此之外，测试还只应在安全的环境中使用已知的、受控的数据库来执行。测试类型有如下几种：

（1）数据和数据库完整性测试

数据库和数据库进程应作为<项目名称>中的子系统来进行测试。在测试这些子系统时，不应将测试对象的用户界面用做数据的接口。对于数据库管理系统（DBMS），还需要进行深入的研究，以确定可以支持以下测试的工具和方法。数据库测试如表 1-8 所示。

表 1-8　数据库测试说明表

测试目标	确保数据库访问方法和进程正常运行，数据不会遭到损坏。
方法	调用各个数据库访问方法和进程，并在其中填充有效的和无效 的数据或对数据的请求。 检查数据库，确保数据已按预期的方式填充，并且所有 数据库事件都按正常方式出现；或者检查所返回的数据，确保为正当的理由检索到了正确的数据。
完成标准	所有的数据库访问方法和进程都按照设计的方式运行，数据没有遭到损坏。
需考虑的特殊事项	测试可能需要 DBMS 开发环境或驱动程序以便在数据库中直接输入或修改数据。 进程应该以手工方式调用。 应使用小型或最小的数据库（其中的记录数很有限）来使所有无法接受的事件具有更大的可见性。

（2）功能测试

测试对象的功能测试应该侧重于可以被直接追踪到用例或业务功能和业务规则的所有测试需求。这些测试的目标在于核实能否正确地接受、处理和检索数据以及业务规则是否正确实施。这种类型的测试基于黑盒方法，即通过图形用户界面（GUI）与应用程序交互并分析输出结果来验证应用程序及其内部进程。表 1-9 列出的是每个应用程序推荐的测试方法概要：

表 1-9　功能测试说明表

测试目标	确保测试对象的功能正常，其中包括导航、数据输入、处理和检索等。
方法	利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以下内容： 在使用有效数据时得到预期的结果。 在使用无效数据时显示相应的错误消息或警告消息。 各业务规则都得到了正确的应用。
完成标准	所计划的测试已全部执行。 所发现的缺陷已全部解决。
需考虑的特殊事项	确定或说明那些将对功能测试的实施和执行造成影响的事项或因素（内部的或外部的）。

（3）业务周期测试

业务周期测试应模拟在一段时间内对<项目名称>执行的活动。应先确定一段时间（例如一年），然后执行将在该时段内发生的事务和活动。这种测试包括所有的每日、每周和每月的周期，以及所有与日期相关的事件（如备忘录）。业务周期测试如表 1-10 所示。

表 1-10　业务周期测试说明表

测试目标	确保测试对象及后台进程都按照所要求的业务模型和时间表正确运行。
方法	通过执行以下活动，测试将模拟若干个业务周期： 将修改或增强对测试对象进行的功能测试，以增加每项功能的执行次数，从而在指定的时段内模拟若干个不同的用户。 将使用有效的和无效的日期或时段来执行所有与时间或日期相关的功能。 将在适当的时候执行或启动所有周期性出现的功能。 在测试中还将使用有效的和无效的数据，以核实以下内容： 在使用有效数据时得到预期的结果。 在使用无效数据时显示相应的错误消息或警告消息。 各业务规则都得到了正确的应用。
完成标准	所计划的测试已全部执行。 所发现的缺陷已全部解决。
需考虑的特殊事项	系统日期和事件可能需要特殊的支持活动。 需要通过业务模型来确定相应的测试需求和测试过程。

（4）用户界面测试

通过用户界面（UI）测试来核实用户与软件的交互。UI 测试的目标在于确保用户界面向用户提供了适当的访问和浏览测试对象功能的操作。除此之外，UI 测试还要确保 UI 功能内部的对象符合预期要求，并遵循公司或行业的标准。用户界面测试如表 1-11 所示。

表 1-11　用户界面测试说明表

测试目标	核实以下内容： 通过浏览测试对象可正确反映业务的功能和需求，这种浏览包括窗口与窗口之间、字段与字段之间的浏览，以及各种访问方法（Tab 键、鼠标移动和快捷键）的使用。 窗口的对象和特征（例如：菜单、大小、位置、状态和中心）都符合标准。

续表

方法	为每个窗口创建或修改测试，以核实各个应用程序窗口和对象都可正确地进行浏览，并处于正常的对象状态。
完成标准	证实各个窗口都与基准版本保持一致，或符合可接受标准。
需考虑的特殊事项	并不是所有定制或第三方对象的特征都可访问。

（5）性能评价

性能评价是一种性能测试，它对响应时间、事务处理速率和其他与时间相关的需求进行评测和评估。性能评价的目标是核实性能需求是否都已满足。实施和执行性能评价的目的是将测试对象的性能为当做条件（例如工作量或硬件配置）的一种函数来进行评价和微调。性能测试如表 1-12 所示。

注：以下事务均指“逻辑业务事务”。这种事务被定义为将由系统的某个主角通过使用测试对象来执行的特定用例，例如，添加或修改某个合同。

表 1-12　性能测试说明表

测试目标	核实所指定的事务或业务功能在以下情况下的性能行为： 正常的预期工作量。 预期的最繁重工作量。
方法	使用为功能或业务周期测试制定的测试过程。 通过修改数据文件来增加事务数量，或通过修改脚本来增加每项事务的迭代次数。 脚本应该在一台计算机上运行（最好是以单个用户、单个事务为基准），并在多台客户机（虚拟的或实际的客户机，请参见下面的“需考虑的特殊事项”）上重复。
完成标准	单个事务或单个用户：在每个事务所预期或要求的时间范围内 成功地完成测试脚本，没有发生任何故障。 多个事务或多个用户：在可接受的时间范围内成功地完成测试 脚本，没有发生任何故障。
需考虑的特殊事项	综合的性能测试还包括在服务器上添加后台工作量。 可采用多种方法来执行此操作，其中包括： 直接将“事务强行分配到”服务器上，这通常以“结构化查询语言”（SQL）调用的形式来实现。 通过创建“虚拟的”用户负载来模拟许多个（通常为数百个）客户机。此负载可通过“远程终端仿真”（Remote Terminal Emulation）工具来实现。此技术还可用于在网络中加载“流量”。 使用多台实际客户机（每台客户机都运行测试脚本）在系统上添加负载。 性能测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。 性能测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。

（6）负载测试

负载测试是一种性能测试。在这种测试中，将使测试对象承担不同的工作量，以评测和评估测试对象在不同工作量条件下的性能行为，以及持续正常运行的能力。负载测试的目标是确定并确保系统在超出最大预期工作量的情况下仍能正常运行。此外，负载测试还要评估性能特征，例如，响应时间、事务处理速率和其他与时间相关的方面。负载测试如表 1-13 所示。

注：以下事务均指“逻辑业务事务”。这些事务被定义为将由系统的最终用户通过使用应用程序来执行的具体功能，例如，添加或修改某个合同。

表 1-13　负载测试说明表

测试目标	核实所指定的事务或商业理由在不同的工作量条件下的性能行为时间。
方法	使用为功能或业务周期测试制定的测试。 通过修改数据文件来增加事务数量，或通过修改测试来增加每项 事务发生的次数。
完成标准	多个事务或多个用户：在可接受的时间范围内成功地完成测试，没有发生任何故障。
需考虑的特殊事项	负载测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。 负载测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。

（7）强度测试

强度测试是一种性能测试，实施和执行此类测试的目的是找出因资源不足或资源争用而导致的错误。如果内存或磁盘空间不足，测试对象就可能会表现出一些在正常条件下并不明显的缺陷。而其他缺陷则可能是由于争用共享资源（如数据库锁或网络带宽）而造成的。强度测试还可用于确定测试对象能够处理的最大工作量。强度测试如表 1-14 所示。注：以下提到的事务都是指逻辑业务事务。

表 1-14　强度测试说明表

测试目标	核实测试对象能够在以下强度条件下正常运行，不会出现任何错误： 服务器上几乎没有或根本没有可用的内存（RAM 和 DASD）。 连接或模拟了最大实际（或实际可承受）数量的客户机。 多个用户对相同的数据/账户执行相同的事务。 最繁重的事务量或最差的事务组合（请参见上面的“性能测试”） 注：强度测试的目标还可表述为确定和记录那些使系统无法继续正常运行的情况或条件。
方法	使用为性能评价或负载测试制定的测试。 要对有限的资源进行测试，就应该在一台计算机上运行测试，而且应该减少或限制服务器上的RAM 和 DASD。 对于其他强度测试，应该使用多台客户机来运行相同的测试或互补的测试，以产生最繁重的事务量或最差的事务组合。
完成标准	所计划的测试已全部执行，并且在达到或超出指定的系统限制时没有出现任何软件故障，或者导致系统出现故障的条件并不在指定的条件范围之内。
需考虑的特殊事项	如果要增加网络工作强度，可能会需要使用网络工具来给网络加载消息或信息包。 应该暂时减少用于系统的 DASD，以限制数据库可用空间的增长。 使多个客户机对相同的记录或数据账户同时进行的访问达到同步。

（8）容量测试

容量测试使测试对象处理大量的数据，以确定是否达到了将使软件发生故障的极限。容量测试还将确定测试对象在给定时间内是否能够持续处理的最大负载或工作量。例如，如果测试对象正在为生成一份报表而处理一组数据库记录，那么容量测试就会使用一个大型的测

试数据库，检验该软件是否正常运行并生成了正确的报表。容量测试如表 1-15 所示。

表 1-15　容量测试说明表

测试目标	核实测试对象在以下大容量条件下能否正常运行： 连接（或模拟了）最大（实际或实际可承受）数量的客户机，所有客户机在长时间内执行相同的、且情况（性能）最差的业务功能。 已达到最大的数据库大小（实际的或按比例缩放的），而且同时执行了多个查询或报表事务。
方法	使用为性能评价或负载测试制定的测试。 应该使用多台客户机来运行相同的测试或互补的测试，以便在长时间内产生最繁重的事务量或最差的事务组合（请参见上面的“强度测试”）。 创建最大的数据库大小（实际的、按比例缩放的、或输入了代表性数据的数据库），并使用多台客户机在长时间内同时运行查询和报表事务。
完成标准	所计划的测试已全部执行，而且在达到或超出指定的系统限制时没有出现任何软件故障。
需考虑的特殊事项	对于上述的大容量条件，哪个时段是可以接受的时间？

（9）安全性和访问控制测试

安全性和访问控制测试侧重于安全性的两个关键方面（如表 1-16 所示）：

a．应用程序级别的安全性，包括对数据或业务功能的访问。

b．系统级别的安全性，包括对系统的登录或远程访问。

应用程序级别的安全性可确保：在预期的安全性情况下，主角只能访问特定的功能或用例，或者只能访问有限的数据。例如，可能会允许所有人输入数据，创建新账户，但只有经理才能删除这些数据或账户。如果具有数据级别的安全性，测试就可确保“用户类型一”能够看到所有客户信息，（包括财务数据），而“用户类型二”只能看见同一客户的统计数据。

系统级别的安全性可确保只有具备系统访问权限的用户才能访问应用程序，而且只能通过相应的网关来访问。

表 1-16　安全性和访问控制测试说明表

测试目标	应用程序级别的安全性：核实主角只能访问其所属用户类型已被授权使用的那些功能或数据。 系统级别的安全性：核实只有具备系统和应用程序访问权限的主角才能访问系统和应用程序。
方法	应用程序级别的安全性：确定并列出各用户类型及其被授权使用的功能或数据。 为各用户类型创建测试，并通过创建各用户类型所特有的事务来核实其权限。 修改用户类型并为相同的用户重新运行测试。对于每种用户类型，确保正确地提供或拒绝了这些附加的功能或数据。 系统级别的访问：请参见下面的“需考虑的特殊事项”。
完成标准	各种已知的主角类型都可访问相应的功能或数据，而且所有事务都按照预期的方式运行，并在先前的应用程序功能测试中运行了所有的事务。
需考虑的特殊事项	必须与相应的网络或系统管理员一起对系统访问权进行检查和讨论。由于此测试可能是网络管理或系统管理的职能，可能不需要执行此测试。

（10）故障转移和恢复测试

故障转移和恢复测试可确保测试对象能成功完成故障转移，并从硬件、软件或网络等方

面的各种故障中进行恢复，这些故障导致数据意外丢失或破坏了数据的完整性。

故障转移测试可确保：对于必须始终保持运行状态的系统来说，如果发生了故障，那么备选或备份的系统就适当地将发生故障的系统“接管”过来，而且不会丢失任何数据或事务。

恢复测试是一种相反的测试流程。其中，将应用程序或系统置于极端的条件下（或者是模仿的极端条件下），以产生故障，例如设备输入/输出（I/O）故障或无效的数据库指针和关键字。启用恢复流程后，将监测和检查应用程序和系统，以核实应用程序或系统是正确无误的，或数据已得到了恢复，如表 1-17 所示。

表 1-17　故障转移和恢复测试说明表

测试目标	确保恢复进程（手工或自动）将数据库、应用程序和系统正确地恢复到了预期的已知状态。测试中将包括以下各种情况： 客户机断电。 服务器断电。 通过网络服务器产生的通信中断。 DASD 和/或 DASD 控制器被中断、断电或与 DASD 和/或 DASD 控制器的通信中断。 周期未完成（数据过滤进程被中断，数据同步进程被中 断）。 数据库指针或关键字无效。 数据库中的数据元素无效或遭到破坏。
方法	应该使用为功能和业务周期测试创建的测试来创建一系列的事务。一旦达到预期的测试起点，就应该分别执行或模拟以下操作： 客户机断电：关闭 PC 的电源。 服务器断电：模拟或启动服务器的断电过程。 通过网络服务器产生的中断：模拟或启动网络的通信中断（实际断开通信线路的连接或关闭网络服务器或路由器的电源）。 DASD 和 DASD 控制器被中断、断电或与 DASD 和 DASD 控制器的通信中断：模拟与一个或多个 DASD 控制器或设备的通信，或实际取消这种通信。 一旦实现了上述情况（或模拟情况），就应该执行其他事务。而且一旦达到第二个测试点状态，就应调用恢复过程。 在测试不完整的周期时，所使用的方法与上述方法相同，只不过应异常终止或提前终止数据库进程本身。 对以下情况的测试需要达到一个已知的数据库状态。当破坏若干个数据库字段、指针和关键字时，应该以手工方式在数据库中（通过数据库工具）直接进行。其他事务应该通过使用“应用程序功能测试”和“业务周期测试”中的测试来执行，并且应执行完整的周期。
完成标准	在所有上述情况中，应用程序、数据库和系统应该在恢复过程完成时立即返回到一个已知的预期状态。此状态包括仅限于已知损坏的字段、指针或关键字范围内的数据损坏，以及表明进程或事务因中断而未被完成的报表。
需考虑的特殊事项	恢复测试会给其他操作带来许多的麻烦。断开缆线连接的方法（模拟断电或通信中断）可能并不可取或不可行。所以，可能会需要采用其他方法，例如诊断性软件工具。 需要系统（或计算机操作）、数据库和网络组中的资源。 这些测试应该在工作时间之外或在一台独立的计算机上运行。

（11）配置测试

配置测试核实测试对象在不同的软件和硬件配置中的运行情况。在大多数生产环境中，客户机工作站、网络连接和数据库服务器的具体硬件规格会有所不同。客户机工作站可能会

安装不同的软件，例如，应用程序、驱动程序等。而且在任何时候，都可能运行许多不同的软件组合，从而占用不同的资源。配置测试如表 1-18 所示。

表 1-18　故障转移和恢复测试说明表

测试目标	核实测试对象可在要求的硬件和软件配置中正常运行。
方法	使用功能测试脚本。 在测试过程中或在测试开始之前，打开各种与非测试对象相关的软件（例如 Microsoft 应用程序：Excel 和 Word），然后将其关闭。 执行所选的事务，以模拟主角与测试对象软件和非测试对象软件之间的交互。 重复上述步骤，尽量减少客户机工作站上的常规可用内存。
完成标准	对于测试对象软件和非测试对象软件的各种组合，所有事务都成功完成，没有出现任何故障。
需考虑的特殊事项	需要可以使用并可以通过桌面访问哪种非测试对象软件？ 通常使用的是哪些应用程序？ 应用程序正在运行什么数据？例如，在 Excel 中打开的大型电子表格，或是在 Word 中打开的 100 页文档。 作为此测试的一部分，应将整个系统、Netware、网络服务器、数据库等都记录下来。

（12）安装测试

安装测试有两个目的。第一个目的是确保该软件能够在所有可能的配置下进行安装，例如，进行首次安装、升级、完整的或自定义的安装，以及在正常和异常情况下安装。异常情况包括磁盘空间不足、缺少目录创建权限等。第二个目的是核实软件在安装后可立即正常运行。这通常是指运行大量为功能测试制定的测试。安装测试说明如表 1-19 所示。

表 1-19　安装测试说明表

测试目标	核实在以下情况下，测试对象可正确地安装到各种所需的硬件配置中： 首次安装。以前从未安装过<项目名称>的新计算机。 以前安装过相同版本的<项目名称>的计算机。 以前安装过较早版本的<项目名称>的计算机。
方法	手工开发脚本或开发自动脚本，以验证目标计算机的状况—新<项目名称>从未安装过；已安装<项目名称>相同或较早版本。 启动或执行安装。 使用预先确定的功能测试脚本子集来运行事务。
完成标准	<项目名称>事务成功执行，没有出现任何故障。
需考虑的特殊事项	应该选择<项目名称>的哪些事务才能准确地测试出<项目名称>应用程序已经成功安装，而且没有遗漏主要的软件构件？

五、工具

此项目将使用表 1-20 所示的工具。注：可以视情况删除或添加项目。

表 1-20　使用工具说明表

	工　　具	厂商/自行研制	版　　本
测试管理			
缺陷跟踪			
用于功能性测试的 ASQ 工具			
用于性能测试的 ASQ 工具			
测试覆盖监测器或评价器			
项目管理			
DBMS 工具			

六、资源

本部分列出推荐<项目名称>项目使用的资源，及其主要职责、知识或技能。

（1）人力资源

表 1-21 列出了在此项目的人员配备方面所做的各种假定。注：可视情况删除或添加项目。

表 1-21　人力资源说明表

人 力 资 源		
角　　色	推荐的最少资源（所分配的专职角色数量）	具体职责或注释
测试组长		负责拟定软件项目的测试计划和方案，提供测试技术指导，组织测试资源，安排测试计划实施，提交测试分析报告，总结整个测试活动。
测试设计员		参与制定测试计划，生成测试模型，在面向对象的设计系统中确定并定义测试类的操作、属性和关联关系，确定测试用例，指导测试实施，参与测试评估和测试分析报告的编写。
测试员		执行实施测试，填写测试记录，记录结果和缺陷。

（2）系统资源

表 1-22 列出了测试项目所需的系统资源。

此时并不完全了解测试系统的具体元素。建议让系统模拟生产环境，并在适当的情况下减小访问量和数据库大小。注：可以视情况删除或添加项目。

表 1-22　系统资源说明表

系 统 资 源	
资　　源	名称/类型
数据库服务器	
网络或子网	
服务器名	
数据库名	
客户端测试 PC	

续表

系 统 资 源	
资　　源	名称/类型
包括特殊的配置需求	
测试存储库	
网络或子网	
服务器名	
测试开发 PC	

七、测试进度和里程碑

（1）项目测试进度

以下测试工作任务的起止时间为：

① 制定测试计划

- 确定测试需求
- 评估风险
- 制定测试策略
- 确定测试资源
- 创建时间表
- 生成测试计划

② 设计测试

- 准备测试计划说明书
- 确定并说明测试用例
- 复审和评估测试覆盖

③ 实施测试

- 单元测试阶段
- 集成测试阶段
- 系统测试阶段
- 提交测试分析报告

④ 测试活动总结

（2）测试里程碑

对<项目名称>的测试应包括上面各节所述的各项测试的测试活动。应该为这些测试确定单独的项目里程碑，如表 1-23 所示，以通知项目的状态和成果。

表 1-23　测试里程碑说明表

里程碑任务	工　作　量	开 始 日 期	结 束 日 期
制定测试计划			
设计测试			
实施测试			
评估测试			

八、可交付工件

这部分内容列出了将要创建的各种文档、工具和报告，及其创建人员、交付对象和交付时间。如：测试计划说明书、测试用例或测试脚本、开发的测试工具、测试日志、缺陷报告、测试分析报告、测试总结等。

（1）概述

① 测试目的

提供一个对项目软件进行测试的总体安排和进度计划，确定现有项目的信息和应测试的软件标明推荐的测试需求（高层次）和可采用的测试策略，并对这些策略加以说明确定所需的资源，并对测试的工作量进行估计，列出测试项目的可交付元素。

② 测试范围

描述测试的各个阶段，例如：单元测试、集成测试或系统测试，并说明本计划所针对的测试类型（如功能测试或性能测试）。简要地列出测试对象中将接受测试或将不接受测试的那些特性和功能。

如果在编写此文档的过程中做出的某些假设可能会影响测试设计、开发或实施，则列出所有这些假设。列出可能会影响测试设计、开发或实施的所有风险或意外事件。列出可能会影响测试设计、开发或实施的所有约束。

③ 限制条件

a．设备所用到的设备类型、数量和预定使用时间。

b．软件列出将被用来支持本项测试过程而本身又并不是被测软件的组成部分的软件，如测试驱动程序、测试监控程序、仿真程序、桩模块等。

c．列出在测试工作期间预期可由用户和开发任务组提供的工作人员的人数。技术水平及有关的预备知识，包括一些特殊要求，如倒班操作和数据输入人员。

④ 参考文档

列出制作此测试计划所依据的文档，如：需求规约、设计规约，概要或详细设计、业务流程、数据流程等。列出要用到的参考资料。

（2）测试摘要

① 测试目标

② 资源和工具

a．资源

项目使用的资源，及其主要职责、知识或技能。

b．工具

列出测试所使用的测试工具或自主开发的测试软件，说明运用这些工具或开发软件测试对象的何种特性。

③ 送测要求

④ 测试种类

（3）测试风险

（4）暂停标准和再启动要求

（5）测试任务和进度

列出要测试中的每一项测试内容，例如：

模块功能测试；

接口正确性测试；

数据文件存取的测试；

运行时间的测试；

设计约束和极限的测试等。

并针对每项测试内容给出测试条件，如：

所用到的设备、数量和预定使用时间；

给出对这项测试的进度安排，包括进行测试的日期和工作内容（如熟悉环境、培训、准备输入数据等）。

（6）测试提交物

① 测试计划

② 测试用例

③ 缺陷记录

④ 测试总结

1.2.3 拓展任务

独立制定天天超市管理系统的测试计划。

第 2 章　测试用例

本章介绍网上购物系统的测试用例的设计与编写。测试用例设计的目的是为每一个测试需求确定测试用例集（Test Case Suite）。

本章重点：

- 测试用例编写规范
- 测试用例设计方法

测试用例是为有效发现软件缺陷而编写的包含测试目的、测试步骤、期望测试结果的特定集合，是测试的基础。设计测试用例是在了解软件业务流程的基础上进行的。设计测试用例的原则是受最小的影响提供最多的测试信息，设计测试用例的目标是一次尽可能的包含多个测试要素。这些测试用例必须是测试工具可以实现的，不同的测试场景将测试不同的功能。

软件的功能和结构越来越复杂，就按最基本的功能测试来说，光靠人脑计算你都测试了什么功能或功能点是不符合实际的也是不可能完成的任务，这样你的测试就没有逻辑性是盲目的。测试用例就是帮助你非常系统地完成测试，并且能帮助你很容易地重现缺陷。编写用例是个很烦琐的工作，包括后面章节中介绍的执行测试也一样，因此耐心是很重要的。

本章工作任务 1 到 4 为功能测试，工作任务 5 为系统测试。

工作任务 2.1　Test Suite用户管理

2.1.1　Test Suite添加注册信息

1．工作任务描述

用户管理是网上购物系统的基本模块，而添加用户注册信息是用户管理模块中的基本功能，也是必需的功能。当用户在浏览器的地址栏中输入本系统的网址时，系统弹出如图 2-1 所示的主页面。

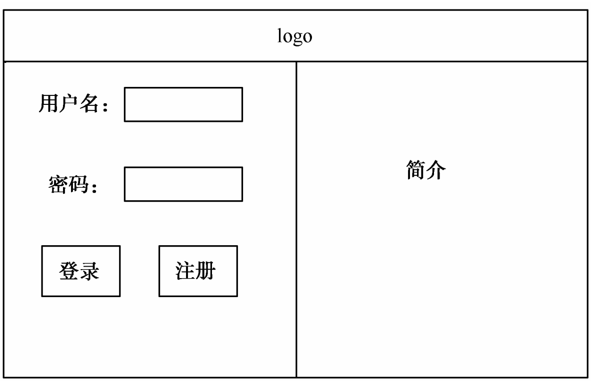


图 2-1　主页面

单击注册按钮，转到如图 2-2 所示的页面中，用户填写用户名、姓名、密码和邮寄地址等信息进行注册，填写完之后单击提交按钮进行注册。如果注册成功则会跳转到如图 2-3 所示的页面。由于系统会对注册信息进行一个简单的验证，如果验证注册信息失败，则系统会提示注册失败信息。

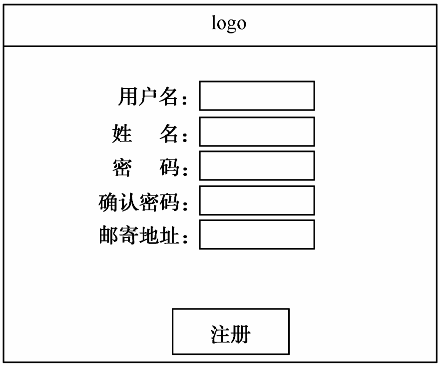


图 2-2　用户注册界面

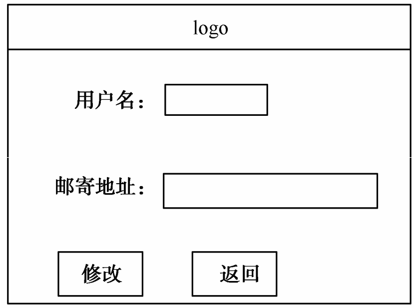


图 2-3　注册成功界面

本节任务就是对添加注册信息功能进行测试，编写测试用例集。在此我们使用了场景法、边界值法、错误推测法等测试用例设计方法。

2．工作过程

编写测试用例集：

以下是用户管理模块中添加注册信息功能的测试用例集。

<table>
<tr><td colspan="2">说明：执行每一步 Steps 时，请参照对应编号的 Expected Results，得出测试结论</td></tr>
<tr><td colspan="2">Test Case 001：必添项是否允许为空</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对必添项为空的情况做了必要的处理</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面中什么都不输入，直接单击［注册］按钮；
4．在“用户名”文本框中输入“小狐狸”后，单击［注册］按钮；
5．重复执行第 4 步骤，
5.1 输入姓名：“王义”，单击［注册］按钮；
5.2 输入密码：111111，单击［注册］按钮；
5.3 输入确认密码：111111，单击［注册］按钮；
5.4 输入联系电话：88888888，单击［注册］按钮；
5.5 输入邮编：131000，单击［注册］按钮；
5.6 输入邮寄地址：职业技术学院，单击［注册］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．系统提示“请输入用户名”；
4．系统提示“请输入姓名”；
5．系统根据用户输入的情况，依次弹出提示信息：
5.1“请输入密码”；
5.2“请输入确认密码”；
5.3“请输入联系电话”；
5.4“请输入邮编”；
5.5“请输入邮寄地址”；
5.6 弹出“注册成功界面”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 002：必添项仅输入空格</td></tr>
<tr><td colspan="2">Summary：
在必添项中仅输入空格，系统是否能够正确处理？</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面的必添项（“用户名”、“姓名”、“密码”、“确认密码”、“联系电话”、“邮编”、“邮寄地址”）中只输入空格，单击［注册］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．提示“用户名”、“姓名”、“联系电话”、“邮编”、“邮寄地址”不能为空</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 003：输入字符数等于域允许的最大字符数</td></tr>
<tr><td colspan="2">Summary：
系统是否对域的输入长度进行了检验</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面输入以下注册信息：
用户名：seven2008111
姓名：seven vilsce
密码：1111111111
确认密码：1111111111
联系电话：1264-0100-888888888-1234
邮编：012345678912
邮寄地址：0123456789012345678901234567890123456789
0123456789012345678901234567890123456789
01234567890123456789
单击［注册］按钮；
4．在“用户注册”界面输入以下注册信息：
用户名：空格+seven2008110+空格
姓名：空格+seven vilsce+空格
密码：1111111111
确认密码：1111111111
联系电话：空格+1264-0100-888888888-1234+空格］
邮编：空格+012345678912+空格
邮寄地址：空格+0123456789012345678901234567890123456789
0123456789012345678901234567890123456789
01234567890123456789+空格
单击［注册］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．系统弹出“注册成功界面”；
4．系统弹出“注册成功界面”</td></tr>
<tr><td colspan="2">边界值法错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 004：输入字符数大于域允许的最大字符数</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对域输入长度进行了验证</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面中将以下信息复制到相应的输入框中：
用户名：seven20081119
姓名：seven vilsce9
密码：11111111119
确认密码：11111111119
联系电话：1264-0100-888888888-12349
邮编：0123456789129
邮寄地址：0123456789012345678901234567890123456789
0123456789012345678901234567890123456789
012345678901234567899
单击［注册］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3.1 预期一：复制信息时，系统自动将信息截断，并弹出提示信息“您输入的信息超长，系统已自动为您截断”；
3.2 预期二：单击［注册］按钮后，系统弹出提示信息“您输入的部分内容已超过系统允许输入的最大字符数，请重新输入”。关闭提示信息后，相关内容已用突出的颜色（如：红色）或者图标（如：“x”）标识出来了</td></tr>
<tr><td colspan="2">边界值法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 005：tab 校验</td></tr>
<tr><td colspan="2">Summary：
检测单击“tab 键”，光标是否能够按照从左至右，由上到下的顺序在输入域间切换</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．将鼠标移动到“用户名”输入框，单击鼠标左键；
4．单击“tab 键”；
5．重复执行第 4 步骤</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．光标定位到“用户名”输入框中；
4．光标跳转到“姓名”输入框中；
5．光标依次跳转到“密码”、“确认密码”、“联系电话”、“邮编”、“邮寄地址”输入框中，最后焦点落到“注册”按钮上</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass / Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 006：用户名中包含空格</td></tr>
<tr><td colspan="2">Summary：
检测系统是否对用户名中的空格做了处理</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面输入以下注册信息：
用户名：“米奇”
姓名：seven vilsce
密码：1111111111
确认密码：1111111111
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮
4．在“用户注册”界面输入以下注册信息：
用户名：米奇
姓名：Anny
密码：1111111111
确认密码：1111111111
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮
5．在“用户注册”界面输入以下注册信息：
用户名：空格+米奇+空格
姓名：Anqe
密码：1111111111
确认密码：1111111111
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．弹出“注册成功界面”；
4．弹出“注册成功界面”；
5．系统提示“该用户名已被使用！”</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 007：特殊字符校验</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对特殊字符做了处理</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面输入以下注册信息：
用户名：小狐' 狸 '
姓名：seven#vilsce
密码：11111<>
确认密码：11111<>
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3.1 预期一：系统提示“您在以下信息：“用户名”、“姓名”、“密码”中包含了系统禁用的特殊字符“ ' 、< 、> 、#”，请修正。
3.2 预期二：弹出“注册成功界面”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 008：密码校验</td></tr>
<tr><td colspan="2">Summary：
检验系统是否做了密码校验</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面输入以下注册信息：
用户名：小熊维尼
姓名：seven vilsce
密码：111111
确认密码：111111+空格
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮
4．在“用户注册”界面输入以下注册信息：
用户名：小熊维尼
姓名：seven vilsce
密码：ddf
确认密码：des
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．提示“您输入的密码和确认密码不一致，请重新输入。”
4．提示“您输入的密码和确认密码不一致，请重新输入。”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 009：“用户名”重名校验</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对“用户名”重名做了处理</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面输入以下注册信息：
用户名：米老鼠
姓名：seven vilsce
密码：1111111111
确认密码：1111111111
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮
4．在“用户注册”界面输入以下注册信息：
用户名：米老鼠
姓名：小米
密码：1111111111
确认密码：1111111111
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．弹出“注册成功界面”
4．提示“该用户名已被使用！”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 010：“姓名”重名校验</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对“姓名”重名做了处理</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面输入以下注册信息：
用户名：米老鼠阿飞
姓名：seven vilsce
密码：1111111111
确认密码：1111111111
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮
4．在“用户注册”界面输入以下注册信息：
用户名：唐老鸭
姓名：seven vilsce
密码：1111111111</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．弹出“注册成功界面”
4．弹出“注册成功界面”</td></tr>
</table>

确认密码：1111111111 联系电话：88888888 邮编：131000 邮寄地址：职业技术学院 单击［注册］按钮	
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 011：回车验证	
Summary： 检验系统是否对 Enter 键进行了处理	
Steps： 单击［Enter］键	Expected Results： 相当于单击了［注册］按钮，将注册信息提交到系统中
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case012：过期校验	
Summary： 检验系统是否做了过期处理	
Steps： 1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮； 2．单击［注册］按钮； 3．在“用户注册”界面输入以下注册信息： 用户名：汤尼 姓名：seven 密码：111111 确认密码：111111 联系电话：88888888 邮编：131000 邮寄地址：职业技术学院 30 分钟后，单击［注册］按钮	Expected Results： 1．弹出“网上购物系统”主页； 2．弹出“用户注册”界面； 3．系统提示“网页已过期”
场景法	
Pass/Fail：	Test Notes：
Author admin	

<table>
<tr><td colspan="2">Test Case 013：密码显示校验</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对密码的显示方式做了处理</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面输入以下注册信息：
用户名：三木
姓名：seven
密码：111111
确认密码：111111</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．密码和确认密码，均未显示明文</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 014：用户名大小写校验</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对用户名的大小写做出了正确的处理</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．单击［注册］按钮；
3．在“用户注册”界面输入以下注册信息：
用户名：abc
姓名：seven vilsce
密码：1111111111
确认密码：1111111111
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮
4．在登录页面中输入
用户名：ABC
密码：1111111111
单击［登录］按钮
5．在“用户注册”界面输入以下注册信息：
用户名：ABC
姓名：seven vilsce
密码：1111111111
确认密码：1111111111
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．系统弹出“注册成功界面”；
4.1 预期一：成功登录到“abc”的个人购物页面。
4.2 预期二：提示“用户名或密码错误，请重新输入”。
5.1 针对 4.1 的预期结果：提示“该用户名已被使用，请使用其他用户名”。
5.2 针对 4.2 的预期结果：弹出“注册成功界面”</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

Test Case 015：页面切换校验	
Summary： 在注册页面和主页之间切换是否正确	
Steps： 1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮； 2．单击［注册］按钮； 3．在“用户注册”界面输入以下注册信息： 用户名：米老鼠蓬勃 姓名：seven vilsce 密码：1111111111 确认密码：1111111111 联系电话：88888888 邮编：131000 邮寄地址：职业技术学院 单击浏览器工具栏上的［后退］按钮 4．单击浏览器工具栏上的［前进］按钮	Expected Results： 1．弹出“网上购物系统”主页； 2．弹出“用户注册”界面； 3．返回到“网上购物系统”主页； 4．进入到“用户注册”界面，密码和确认密码输入域已被清空，其他输入域的信息仍然被保留
场景法	
Pass/Fail：	Test Notes：
Author admin	

2.1.2　Test Suite管理员登录

1．工作任务描述

在本系统中，管理员可以对商品信息和商品的类别信息进行管理。管理员登录界面如图 2-4 所示，当管理员成功登录后，则进入后台管理主界面如图 2-5 所示。

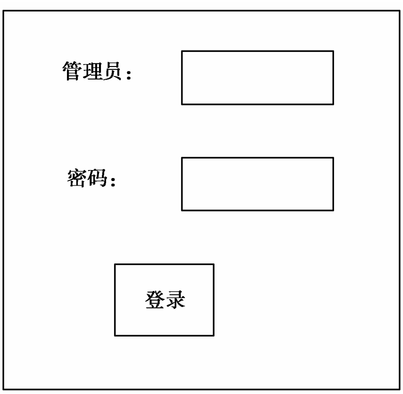


图 2-4　登录界面

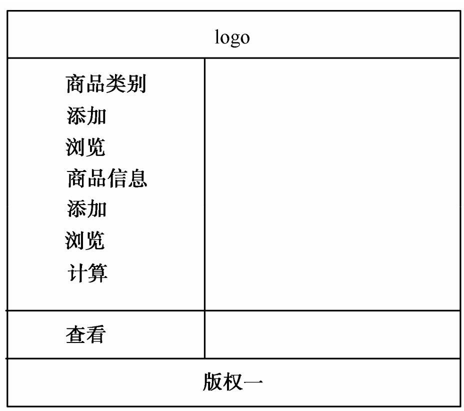


图 2-5　后台管理主界面

本节任务就是对管理员登录功能进行测试，编写测试用例集。在此我们使用了场景法、错误推测法等测试用例设计方法。

2．工作过程

编写测试用例集。

以下是用户管理模块的子功能管理员登录的测试用例集。

<table>
<tr><td colspan="2">Test Case 016：回车验证</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对 Enter 键进行了处理</td></tr>
<tr><td>Steps：
单击［Enter］键</td><td>Expected Results：
相当于单击了［登录］按钮</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 017：登录次数校验</td></tr>
<tr><td colspan="2">Summary：
检验超过登录限制次数后，是否还可以继续登录</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的url，单击［转到］按钮；
2．在管理员登录区中输入
管理员用户名：admin
密码：错误的密码
3．连续数错 5 次，第 6 次登录</td><td>Expected Results：
1．弹出“网上购物系统”主页
2．提示“用户名或密码错”
3．提示“您错误登录次数超限，账户已被锁定！”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 018：权限校验</td></tr>
<tr><td colspan="2">Summary：
检验管理员登录后，是否可以访问被授权的页面</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的url，单击［转到］按钮；
2．在管理员登录区输入
管理员：admin
密码：admin
单击［登录］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页
2．进入后台管理页面，可以维护商品类别、商品、订单信息</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 019：注入式登录</td></tr>
<tr><td colspan="2">Summary：
利用 sql 漏洞,使用不存在的用户登录</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．在管理员登录界面中输入以下信息：
用户名：admin'OR'1'='1
密码：x'OR'1'='1
单击［登录］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．提示“用户名或密码不正确，请重新输入”</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 020：注册用户登录</td></tr>
<tr><td colspan="2">Summary：
在管理员登录区，输入合法的注册用户名和密码</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．在管理员登录区中输入系统中合法的注册用户名和密码
用户名：小狐狸
密码：111111</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2.1 预期一：提示“请到客户区登录”；
2.2 预期二：登录到“狐狸”个人购物主页</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 021：锁定用户登录</td></tr>
<tr><td colspan="2">Summary：
检验锁定用户是否可以登录</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．在管理员登录区中输入
已被锁定的管理员用户名：admin
正确的密码：admin
单击［登录］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．提示“账户已被锁定，请联系技术人员”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

Test Case 022：ab 校验	
Summary： 检测单击“Tab 键”光标是否能够按照从左至右，由上到下的顺序在输入域间切换	
Steps： 1．在浏览器的地址栏中输入访问“网上购物系统”url，单击［转到］按钮； 2．在管理员登录区将鼠标移动到“用户名”输入框，单击鼠标左键； 3．单击“Tab 键”； 4．单击“Tab 键”	Expected Results： 1．弹出“网上购物系统”主页； 2．光标定位到“用户名”输入框中； 3．光标跳转到“密码”输入框中； 4．焦点到“登录”按钮上
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

2.1.3 Test Suite注册用户登录

1．工作任务描述

用户注册成功后，就可以登录网站了，用户登录的页面如图 2-6 所示。登录成功后进入商品购买主界面如图 2-7 所示。

本节任务就是编写已注册过的用户登录功能的测试用例集。在此我们使用了场景法、错误推测法、边界值法等测试用例设计方法。

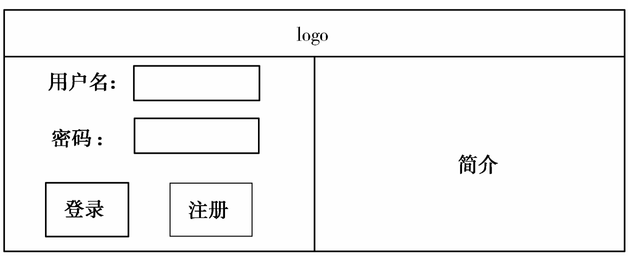


图 2-6　主页面

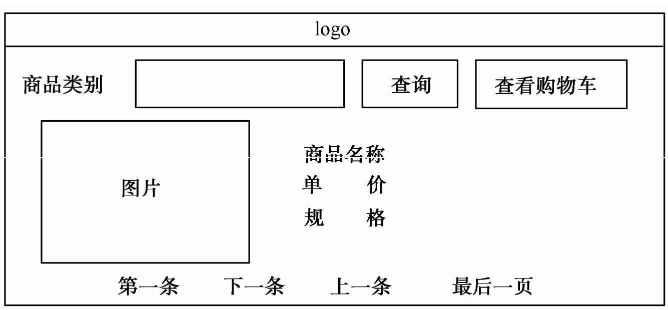


图 2-7　商品购买主界面

2．工作过程

编写测试用例集。

以下是注册用户登录的测试用例集。

<table>
<tr><td colspan="2">Test Case 023：回车验证</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对 Enter 键进行了处理</td></tr>
<tr><td>Steps：
单击［Enter］键</td><td>Expected Results：
相当于单击了［登录］按钮</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 024：登录密码中包含空格</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对密码中的空格做了处理</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．（构造符合条件的测试用户）单击［注册］按钮；
3．在“用户注册”界面输入以下注册信息：
用户名：密码含空格
姓名：seven vilsce
密码：1111+2 个空格
确认密码：1111+2 个空格
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮
4．在注册用户的登录页面上输入
用户名：密码含空格
密码：1111+2 个空格
单击［登录］按钮
5．在登录页面输入
用户名：密码含空格
密码：1111
单击【登录】按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3．系统弹出“注册成功界面”；
4．可以登录到“密码含空格”的个人购物页面；
5．提示“用户名或密码错误，请重新输入”</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 025：登录密码大小写校验</td></tr>
<tr><td colspan="2">Summary：
检验密码校验是否处理了大小写问题</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的url，单击［转到］按钮；
2．（构造符合条件的测试用户）单击［注册］按钮；
3．在“用户注册”界面输入以下注册信息：
用户名：狮子王
姓名：seven vilsce
密码：ABC
确认密码：abc
联系电话：88888888
邮编：131000
邮寄地址：职业技术学院
单击［注册］按钮
4．针对 3.1 的预期结果，修改注册信息
密码：abc
确认密码：abc
单击［注册］按钮
5．在“登录”界面输入以下信息：
用户名：狮子王
密码：ABC
单击［登录］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．弹出“用户注册”界面；
3.1 预期一：提示“您输入的密码和确认密码不一致，请重新输入。”
3.2 预期二：弹出“注册成功界面”；
4．弹出“注册成功界面”；
5．针对 3.1 的预期结果：提示“用户名或密码不正确”；
针对 3.2 的预期结果：成功登录到“狮子王”的个人购物页面</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 026：登录次数校验</td></tr>
<tr><td colspan="2">Summary：
检验超过登录限制次数后，是否还可以继续登录</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的url，单击［转到］按钮；
2．在登录界面中输入：
用户名：唐老鸭
密码：错误的密码
3．连续数错 5 次，第 6 次登录</td><td>Expected Results：
1．弹出“网上购物系统”主页
2．提示“用户名或密码错”
3．提示“您错误登录次数超限，账户已被锁定！”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 027：使用字符长度等于临界值的用户名和密码登录</td></tr>
<tr><td colspan="2">Summary：
使用信息长度等于域允许的最大长度的用户名和密码登录</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的url，单击［转到］按钮；
2．在登录窗口中输入系统中存在的
用户名：seven2008111
密码：1111111111
单击［登录］按钮
3．单击［注销］退出个人购物窗口
在登录窗口中输入
用户名：空格+seven2008111+空格
密码：1111111111
单击［登录］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．进入“seven2008111”的个人购物主页面；
3．仍然可以进入“seven2008111”的个人购物主页面</td></tr>
<tr><td colspan="2">边界值法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 028：使用含有空格的用户名登录</td></tr>
<tr><td colspan="2">Summary：
使用用户名内含有空格的用户名登录</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的url，单击［转到］按钮；
2．在登录窗口中输入
用户名：米奇
密码：1111111111
单击［登录］按钮
3．单击［注销］退出个人购物窗口；在登录窗口中输入
用户名：空格+米奇+空格
密码：1111111111
单击［登录］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．进入“米奇”的个人购物主页面；
3．仍然可以进入“米奇”的个人购物主页面</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 029：使用超长用户名和密码登录</td></tr>
<tr><td colspan="2">Summary：
在登录页面的用户名和密码输入框中输入系统允许最大长度的用户名和密码</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．在登陆界面中输入以下信息：
用户名：aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
密码：111111111111111111111111111111111111111111111</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．提示“用户名或密码不正确，请重新输入”，而不应该出现错误页</td></tr>
<tr><td colspan="2">边界值法错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 030：注入式登录</td></tr>
<tr><td colspan="2">Summary：
利用 sql 漏洞，使用不存在的用户登录</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．在登陆界面中输入以下信息：
用户名：x'OR'1'='1
密码：x'OR'1'='1
单击［登录］按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页；
2．提示“用户名或密码不正确，请重新输入”</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 031：锁定用户登录</td></tr>
<tr><td colspan="2">Summary：
检验锁定用户是否可以登录</td></tr>
<tr><td>Steps：
1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮；
2．在前台客户登录区中输入
已被锁定的用户名：唐老鸭
正确的密码：1111111111
单击【登录】按钮</td><td>Expected Results：
1．弹出“网上购物系统”主页
2．提示“账户已被锁定，请联系技术人员”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

Test Case 032：Tab 校验	
Summary： 检测单击“Tab 键”，光标是否能够按照从左至右，由上到下的顺序在输入域间切换	
Steps： 1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮； 2．在用户登录区将鼠标移动到“用户名”输入框，单击鼠标左键； 3．单击“Tab 键”； 4．单击“Tab 键”	Expected Results： 1．弹出“网上购物系统”主页； 2．光标定位到“用户名”输入框中； 3．光标跳转到“密码”输入框中； 4．焦点到“登录”按钮上
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

2.1.4 Test Suite修改注册信息

1．工作任务描述

用户成功登录系统后，可以对自己的信息进行修改。修改注册信息的界面如图 2-8 所示。

本节任务就是编写修改注册信息功能的测试用例集。在此我们使用了场景法、错误推测法、边界值法等测试用例设计方法。

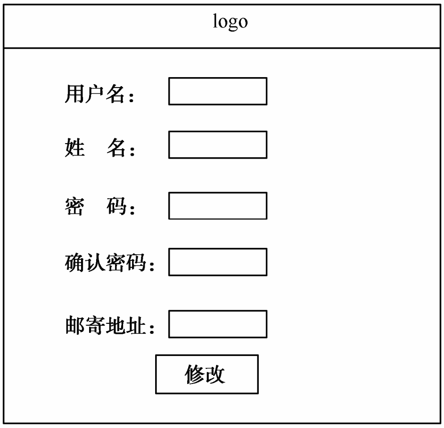


图 2-8　修改注册信息

2．工作过程

编写测试用例集。

以下是修改注册信息的测试用例集。

<table>
<tr><td colspan="2">Test Case 033：必添项是否允许为空</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对必填项为空的情况做了处理</td></tr>
<tr><td>Steps：
1．输入用户名：米奇，密码：1111111111，单击［登录］按钮；
2．单击［修改个人信息］按钮；
3．在“修改个人信息”界面中将“姓名”信息删除；单击［保存］按钮；
4．在“姓名”文本框中输入“小大大”后，单击［保存］按钮；
5．重复执行第 3 和第 4 步骤，将必填项依次删除</td><td>Expected Results：
1．弹出“米奇”个人购物主页；
2．弹出“修改个人信息”界面；
3．系统提示“姓名不允许为空”；
4．提示“修改成功”；
5．提示“必添项不允许为空”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 034：必添项仅输入空格</td></tr>
<tr><td colspan="2">Summary：
在必添项中仅输入空格，系统是否能够正确处理</td></tr>
<tr><td>Steps：
1．输入用户名：米奇，密码：1111111111，单击［登录］按钮；
2．单击［修改个人信息］按钮；
3．在“修改个人信息”界面的必添项（“用户名”、“姓名”、“密码”、“确认密码”、“联系电话”、“邮编”、“邮寄地址”）中只输入空格，单击［保存］按钮</td><td>Expected Results：
1．弹出“米奇”个人购物主页；
2．“弹出修改个人信息“界面；
3．提示“用户名”、“姓名”、“联系电话”、“邮编”、“邮寄地址”不能为空</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 035：输入字符数大于域允许的最大字符数</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对输入域的长度进行了验证</td></tr>
<tr><td>Steps：
1．输入用户名：seven2008111，密码：1111111111，单击［登录］按钮；
2．单击［修改个人信息］按钮；
3．在“修改个人信息”界面中将以下信息复制到相应的输入框中：
用户名：seven20081119
姓名：seven vilsce9
密码：11111111119</td><td>Expected Results：
1．进入“seven2008111”的个人购物主页；
2．弹出“修改个人信息”界面；
3.1 预期一：复制信息时，系统自动将信息截断，并弹出提示信息“您输入的信息超长，系统已自动为您截断”；
3.2 预期二：单击［保存］按钮后，系统弹出提示信息“您输入的部分内容已超过系统允许输入的最大字符数，请重新输入”。关闭提示信息后，相关内容已用突出的颜色（红色）或者图标（x）标识出来了</td></tr>
</table>

确认密码：11111111119 联系电话：1264-0100-888888888-12349 邮编：0123456789129 邮寄地址：0123456789012345678901234567890123456789 0123456789012345678901234567890123456789 01234567890123456789 9 单击［保存］按钮；	
边界值法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 036：输入字符数等于域允许的最大字符数	
Summary： 检验系统是否对输入域的长度进行了验证	
Steps： 1．输入用户名：米老鼠，密码：1111111111，单击［登录］按钮； 2．单击［修改个人信息］按钮； 3．将个人信息修改为如下内容： 用户名：seven2008118 姓名：seven vilsll 密码：1111111188 确认密码：1111111188 联系电话：1264-0100-888888888-8888 邮编：012345678918 邮寄地址：0123456789012345678901234567890123456789 0123456789012345678901234567890123456789 01234567890123456788 单击［保存］按钮； 4．重新打开“修改个人信息”页面，将个人信息修改为如下内容： 用户名：空格+seven2008166+空格 姓名：空格+seven vil222+空格 密码：1111111166 确认密码：1111111166 联系电话：空格+1264-0100-888888888-1666+空格］ 邮编：空格+012345678966+空格 邮寄地址：空格+0123456789012345678901234567890123456789 0123456789012345678901234567890123456789 01234567890123456766+空格 单击［保存］按钮	Expected Results： 1．登录到“米老鼠”的个人购物主页； 2．弹出“修改个人信息”界面； 3．提示“保存成功”。 4．提示“保存成功”
边界值法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 037：Tab 校验	
Summary： 检测单击“Tab 键”，光标是否能够按照从左至右，由上到下的顺序在输入域间切换	
Steps： 1．输入用户名：米奇，密码：1111111111，单击［登录］按钮； 2．单击［修改个人信息］按钮； 3．将鼠标移动到“用户名”输入框，单击鼠标左键； 4．单击“Tab 键”； 5．重复执行第 4 步骤	Expected Results： 1．弹出“米奇”个人购物主页； 2．弹出“修改个人信息”界面； 3．光标已定位到“用户名”输入框中； 4．光标跳转到“姓名”输入框中； 5．光标依次跳转到“密码”、“确认密码”、“联系电话”、“邮编”、“邮寄地址”输入框中，最后焦点落到“注册”按钮上
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 038：用户名中包含空格	
Summary： 检验系统是否对用户名中的空格做出了处理	
Steps： 1．输入用户名：三木，密码：111111，单击［登录］按钮； 2．单击［修改个人信息］按钮； 3．在“修改个人信息”界面中将用户名改为“三木”单击［保存］按钮 4．在“修改个人信息”界面中将用户名改为“米奇+空格”单击［保存］按钮	Expected Results： 1．弹出“三木”个人购物主页； 2．弹出“修改个人信息”界面； 3．提示“保存成功” 4．提示“该用户名已被使用！”
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 039：不修改直接保存	
Summary： 不修改，直接单击［保存］按钮	
Steps： 1．输入用户名：米奇，密码：1111111111，单击［登录］按钮； 2．单击［修改个人信息］按钮； 3．不做任何改动，直接单击［保存］按钮	Expected Results： 1．弹出“米奇”个人购物主页； 2．弹出“修改个人信息”界面； 3．提示“保存成功”（不应该提示“用户名已存在”）
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 040：特殊字符校验	
Summary： 检验系统是否对特殊字符检验做出了正确的处理	
Steps： 1．输入用户名：三木，密码：111111，单击［登录］按钮； 2．单击［修改个人信息］按钮； 3．在“修改个人信息”界面中，将个人信息修改为以下内容： 用户名：小狐' 狸 ' 姓名：seven#vilsce 输入密码：11111<> 确认密码：11111<> 输入联系电话：88888888 邮编：131000 邮寄地址：职业技术学院 单击［保存］按钮	Expected Results： 1．弹出“三木”个人购物主页； 2．弹出“修改个人信息”界面； 3.1 预期一：提示“您在以下信息：“用户名”、“姓名”、“密码”中包含了系统禁用的特殊字符“ ' 、< 、> 、 #”，请修正。” 3.2 预期二：提示“保存成功”
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 041：密码校验	
Summary： 系统是否做了密码校验	
Steps： 1．输入用户名：三木，密码：111111，单击［登录］按钮； 2．单击［修改个人信息］按钮； 3．将密码改为：abc，确认密码改为 dde，单击［保存］按钮； 4．将密码改为：abc，确认密码改为 abc+空格，单击［保存］按钮；	Expected Results： 1．弹出“三木”个人购物主页； 2．弹出“修改个人信息”界面； 3．系统提示“您输入的密码和确认密码不一致，请重新输入。” 4．系统提示“您输入的密码和确认密码不一致，请重新输入。”
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 042：用户名重名校验	
Summary：	
Steps： 1．输入用户名：三木，密码：111111，单击［登录］按钮； 2．单击［修改个人信息］按钮； 3．将用户名改为：米老鼠，单击【保存】按钮；	Expected Results： 1．弹出“三木”个人购物主页； 2．弹出“修改个人信息”界面； 3．系统提示“该用户名已被使用！”
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 043：回车验证	
Summary： 检验系统是否对 Enter 键进行了处理	
Steps：单击［Enter］键	Expected Results：相当于单击了［修改］按钮
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 044：页面切换校验	
Summary： 在修改个人信息页面和主页之间切换是否正确	
Steps： 1．输入用户名：三木，密码：111111，单击［登录］按钮； 2．单击［修改个人信息］按钮； 3．修改密码为：abd，确认密码为：abd 单击浏览器工具栏上的［后退］按钮 4．单击浏览器工具栏上的［前进］按钮	Expected Results： 1．弹出“三木”的个人购物主页； 2．弹出“修改个人信息”界面； 3．返回到“网上购物系统”主页； 4．进入到“修改个人信息”界面，密码和确认密码输入域已被清空，其他输入域的信息仍然被保留
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 045：过期校验	
Summary： 检验系统是否做了过期处理。	
Steps： 1．输入用户名：三木，密码：111111，单击［登录］按钮； 2．单击［修改个人信息］按钮； 3．30 分钟后，单击［保存］按钮	Expected Results： 1．弹出“三木”个人购物主页； 2．弹出“修改个人信息”界面； 3．系统提示“网页已过期”
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 046：密码显示校验	
Summary： 检验系统是否为密码显示进行了特殊处理	
Steps： 1．输入用户名：三木，密码：111111，单击［登录］按钮； 2．单击［修改个人信息］按钮； 3．修改密码和确认密码 姓名：seven 输入密码：111111 确认密码：111111	Expected Results： 1．弹出“三木”个人购物主页； 2．弹出“修改个人信息”界面；密码和确认密码不应以明文方式显示 3．密码和确认密码不应以明文方式显示
场景法	
Pass/Fail：	Test Notes：
Author admin	

2.1.5　应知应会

1．什么是测试用例

测试用例（Test Case）是按一定的顺序执行的并与测试目标相关的测试活动的描述，它确定“怎样”测试。测试用例是有效发现软件缺陷的最小测试执行单元，是软件的测试规格说明书。目前也没有测试用例这个词汇的经典定义，常见的说法是：指对一项特定的软件产品进行测试任务的描述，体现测试方案、方法、技术和策略，内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等，并形成文档。

在测试工作中，测试用例的设计是非常重要的，是测试执行的正确性、有效性的基础。如何有效地设计测试用例一直是测试人员所关注的问题，设计好的测试用例是保证测试工作的最关键的因素之一。

如果问测试工程师测试用例如何编写，就像是问程序员如何编写代码得到的答案一样，每 个人都会给出不同的编写方法，但实用的测试用例却像优秀的程序一样难以编写。目前在国内，测试工程师却时常要面对“已经延期几倍计划时间的项目”，测试用例如何发挥更大的作用，是一个迫切需要解决的问题。事实上，完全可以把测试用例看成是测试工程师编写的程序：这个“程序”是为了辅助测试 工作的进行而开发的，目的是为了发现软件问题，同时“顺便”证明软件功能是否符合要求。

测试用例主要用在集成测试、系统测试以及回归测试中。测试人员按照已规定好的测试用例实施测试，而不得做随意的变动。因为测试用例是分测试等级的，集成测试应测试哪些用例，系统测试应包含哪些用例，以及回归测试又该实施什么样的测试用例，在设计测试用例时都是由专门人员明确规定并形成文档的。在实施测试时测试用例作为测试的标准，测试人员一定要按照测试用例严格按用例项目和测试步骤逐一实施测试，并且要把测试结果详细记录下来，以便形成测试结果文档，等下一轮测试时作为参考之用。

在本书给出的测试用例中，测试数据是给定的，但是有时候在实践当中，测试数据与测试用例是分离的，根据测试用例设计，需要准备大量原始数据及标准测试期望的结果，这些数据包括各种情况下的输入数据尤其是必须设计出大量的边缘数据和错误数据，这些设计用例的方法在后续内容中会有详细介绍。

当测试实施完成后，对测试结果的评估是非常重要的。要判断软件测试是否完成，衡量测试质量需要一些量化的结果（测试覆盖率多少、测试合格率多少、重要测试合格率多少……），这样把测试用例及实施结果作为度量标准使测试更加准确有效。通过实施测试用例，将系统缺陷（Bug）尽量收集全面，把测试用例和缺陷数据进行对比，分析是漏测还是缺陷复现，最终使系统逐步完善软件质量。

当然，测试用例本身在形成文档后也是在不断修改更新与完善，原因有三个：第一，在测试过程中发现设计测试用例时考虑不周，需要完善；第二，在软件交付使用后反馈的软件缺陷，而缺陷又是因测试用例存在漏洞造成的；第三，软件自身的新增功能以及软件版本的更新，测试用例也必须配套修改更新。一些小的修改直接在原测试用例文档里更正就可以了，但是文档中要有此次更正的记录。软件的版本是会升级更新的，测试用例也一样，随着软件的升级而编制新的版本。

测试用例的形成可以分为简单的七个步骤：

1．理清模块需求

2．提出测试需求

3．设计测试思路

4．测试用例编写

5．测试用例评审

6．执行用例

7．用例效率计算

在本书后续应知应会中会有详细用例编写的介绍，在此不再赘述。

2．测试的阶段和种类

性能测试、压力测试、负载测试、强度测试、稳定性测试、健壮性测试、功能测试、接口测试……，这么多眼花缭乱的测试类型名称，估计很少有人能准确地区分并说出定义来，至于对应的 测试用例如何编写和执行，就更不用说了。这里给读者做一下较为系统的说明。

（1）软件测试阶段

软件的测试阶段就是测试将按照什么样的思路和方式进行。通常，软件测试要经过单元测试、集成测试、确认测试、系统测试以及验收测试。这些阶段和开发过程是相对应的。关于测试阶段的叫法有多种，比如测试类型，测试策略等。

① 单元测试：单元测试是针对软件设计的最小单位——程序模块甚至代码段进行正确性检验的测试工作，通常由开发人员进行。

② 集成测试：集成测试是将模块按照设计要求组装起来进行测试，主要目的是发现与接口有关的问题。由于在产品提交到测试部门前，产品开发小组都要进行联合调试，因此在大部分企业中集成测试是由开发人员来完成的。

时常有这样的情况发生，每个模块都能单独工作，但这些模块集成在一起之后却不能正常工作。主要原因是，模块相互调用时接口会引入许多新问题。例如，数据经过接口可能丢失；一个模块对另一模块可能造成不应有的影响；几个子功能组合起来不能实现主功能；误差不断积累达到不可接受的程度；全局数据结构出现错误，等等。综合测试是组装软件的系统测试技术，按设计要求把通过单元测试的各个模块组装在一起之后，进行综合测试以便发现与接口有关的各种错误。

某设计人员习惯于把所有模块按设计要求一次全部组装起来，然后进行整体测试，这称为非增量式集成。这种方法容易出现混乱。因为测试时可能发现一大堆错误，为每个错误定位和纠正非常困难，并且在改正一个错误的同时又可能引入新的错误，新旧错误混杂，更难断定出错的原因和位置。与之相反的是增量式集成方法，程序一段一段地扩展，测试的范围一步一步地增大，错误易于定位和纠正，界面的测试亦可做到完全彻底。下面讨论两种增量式集成方法。

a．自顶向下集成

自顶向下集成是构造程序结构的一种增量式方式，它从主控模块开始，按照软件的控制层次结构，以深度优先或广度优先的策略，逐步把各个模块集成在一起。深度优先策略首先是把主控制路径上的模块集成在一起，至于选择哪一条路径作为主控制路径，这多少带有随意性，一般根据问题的特性确定。以图 2-9 为例，若选择了最左一条路径，首先将模块 M1，M2，M5 和 M8 集成在一起，再将 M6 集成起来，然后考虑中间和右边的路径。广度优先策略则不然，它沿控制层次结构水平地向下移动。仍然以图 2-9 为例，它首先把 M2、M3 和 M4 与主控模块集成在一起，再将 M5 和 M6 和其他模块集成起来。

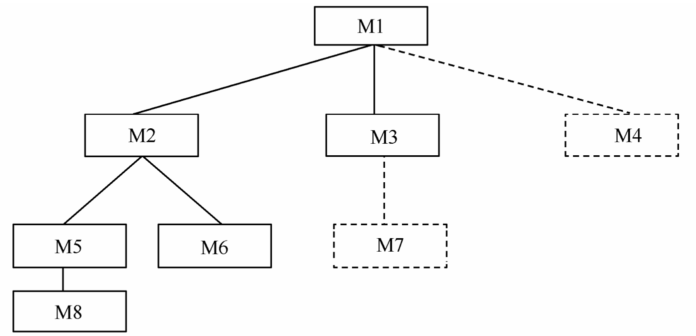


图 2-9　自顶向下集成

由于下文用到桩模块的概念，在此先向读者介绍一下。在集成测试前要为被测模块编制一些模拟其下级模块功能的“替身”模块，以代替被测模块的接口，接受或传递被测模块的数据，这些专供测试用的“假”模块称为被测模块的桩模块。

自顶向下综合测试的具体步骤为：

- 以主控模块作为测试驱动模块，把对主控模块进行单元测试时引入的所有桩模块用实际模块替代；
- 依据所选的集成策略（深度优先或广度优先），每次只替代一个桩模块；
- 每集成一个模块立即测试一遍；
- 只有每组测试完成后，才着手替换下一个桩模块；
- 为避免引入新错误，须不断地进行回归测试（即全部或部分地重复已做过的测试）。

从第二步开始，循环执行上述步骤，直至整个程序结构构造完毕。图 2-9 中，实线表示已部分完成的结构，若采用深度优先策略，下一步将用模块 M7 替换桩模块 S7，当然 M7 本身可能又带有桩模块，随后将被对应的实际模块替代。最后直至桩模块 S4 被替代完毕为止。

自顶向下集成的优点在于能尽早地对程序的主要控制和决策机制进行检验，因此较早地发现错误。缺点是在测试较高层模块时，低层处理采用桩模块替代，不能反映真实情况，重要数据不能及时回送到上层模块，因此测试并不充分。解决这个问题有几种办法，第一种是把某些测试推迟到用真实模块替代桩模块之后进行，第二种是开发能模拟真实模块的桩模块；第三种是自底向上集成模块。第一种方法又回退为非增量式的集成方法，使错误难于定位和纠正，并且失去了在组装模块时进行一些特定测试的可能性；第二种方法无疑要大大增加开销；第三种方法比较切实可行，下面专门讨论。

b．自底向上集成

自底向上测试是从“原子”模块（即软件结构最低层的模块）开始组装测试，因测试到较高层模块时，所需的下层模块功能均已具备，所以不再需要桩模块。

自底向上综合测试的步骤分为：

- 把底层模块组织成实现某个子功能的模块群（cluster）；
- 开发一个测试驱动模块，控制测试数据的输入和测试结果的输出；
- 对每个模块群进行测试；
- 删除测试使用的驱动模块，用较高层模块把模块群组织成为完成更大功能的新模块群。

从第一步开始循环执行上述各步骤，直至整个程序构造完毕。

图 2-10 说明了上述过程。首先“原子”模块被分为三个模块群，每个模块群引入一个驱动模块进行测试。因模块群 1、模块群 2 中的模块均隶属于模块 Ma，因此在驱动模块 D1、D2 去掉后，模块群 1 与模块群 2 直接与 Ma 接口，这时可将 D3 去掉，将 Mb 与模块群 3 直接接口，对 Mb 进行集成测试。这样继续下去，直至最后将驱动模块 D4、D5 也去掉，最后 Ma、Mb 和 Mc 全部集成在一起进行测试。

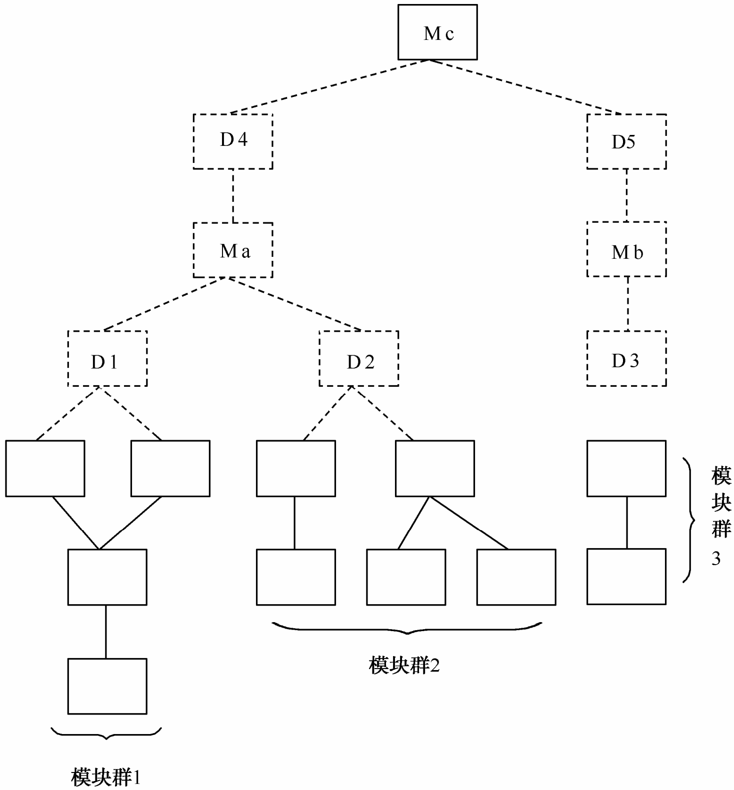


图 2-10　自顶向下集成

自底向上集成方法不用桩模块，测试用例的设计亦相对简单，但缺点是程序最后一个模块加入时才具有整体形象。它与自顶向下综合测试方法优缺点正好相反。因此，在测试软件系统时，应根据软件的特点和工程的进度，选用适当的测试策略，有时混和使用两种策略更为有效，上层模块用自顶向下的方法，下层模块用自底向上的方法。

此外，在综合测试中尤其要注意关键模块，所谓关键模块一般都具有下述一或多个特征：

- 对应几条需求；
- 具有高层控制功能；
- 复杂、易出错；
- 有特殊的性能要求。关键模块应尽早测试，并反复进行回归测试。

③ 系统测试：系统测试是在集成测试通过后进行的，目的是充分运行系统，验证各子系统是否都能正常工作并完成设计的要求。它主要由测试部门进行，是测试部门最大最重要的一个测试，对产品的质量有重大的影响。

计算机软件是基于计算机系统的一个重要组成部分，软件开发完毕后应与系统中其他成分集成在一起，此时需要进行一系列系统集成和确认测试。对这些测试的详细讨论已超出软件工程的范围，这些测试也不可能仅由软件开发人员完成。在系统测试之前，软件工程师应完成下列工作：

- 为测试软件系统的输入信息设计出错处理通路；
- 设计测试用例，模拟错误数据和软件界面可能发生的错误，记录测试结果，为系统测试提供经验和帮助；
- 参与系统测试的规划和设计，保证软件测试的合理性。

系统测试应该由若干个不同测试组成，目的是充分运行系统，验证系统各部件是否都能正常工作并完成所赋予的任务。下面简单讨论几类系统测试。

a．恢复测试

恢复测试主要检查系统的容错能力。当系统出错时，能否在指定时间间隔内修正错误并重新启动系统。恢复测试首先要采用各种办法强迫系统失败，然后验证系统是否能尽快恢复。对于自动恢复需验证重新初始化（reinitialization）、检查点（checkpointing mechanisms）、数据恢复（data recovery）和重新启动（restart）等机制的正确性；对于人工干预的恢复系统，还需估测平均修复时间，确定其是否在可接受的范围内。

b．安全测试

安全测试检查系统对非法侵入的防范能力。安全测试期间，测试人员假扮非法入侵者，采用各种办法试图突破防线。例如，① 想方设法截取或破译口令；② 专门定做软件破坏系统的保护机制；③ 故意导致系统失败，企图趁恢复之机非法进入；④ 试图通过浏览非保密数据，推导所需信息，等等。理论上讲，只要有足够的时间和资源，没有不可进入的系统。因此系统安全设计的准则是，使非法侵入的代价超过被保护信息的价值。此时非法侵入者已无利可图。

c．强度测试

强度测试检查程序对异常情况的抵抗能力。强度测试总是迫使系统在异常的资源配置下运行。例如，① 当中断的正常频率为每秒一至两个时，运行每秒产生十个中断的测试用例；② 定量地增长数据输入率，检查输入子功能的反应能力；③ 运行需要最大存储空间（或其他资源）的测试用例；④ 运行可能导致操作系统崩溃或磁盘数据剧烈抖动的测试用例，等等。

d．性能测试

对于那些实时和嵌入式系统，软件部分即使满足功能要求，也未必能够满足性能要求，虽然从单元测试起，每一测试步骤都包含性能测试，但只有当系统真正集成之后，在真实环境中才能全面、可靠地测试运行操作系统，性能测试就是为了完成这一任务。性能测试有时与强度测试相结合，经常需要其他软硬件的配套支持。

④ 验收测试：验收测试（acceptance testing）以需求阶段的《需求规格说明书》为验收标准，测试时要求模拟实际用户的运行环境。对于实际项目可以和客户共同进行，对于产品来说就是最后一次的系统测试。测试内容为对功能模块的全面测试，尤其要进行文档测试。

验收测试是系统开发生命周期方法论的一个阶段，这时相关的用户和/或独立测试人员根据测试计划和结果对系统进行测试和接收。它让系统用户决定是否接收系统。它是一项确定产品是否能够满足合同或用户所规定需求的测试。这是管理性和防御性控制。

验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪，

并且可以让最终用户将其用于执行软件的既定功能和任务。

验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所期待的那样。

通过综合测试之后，软件已完全组装起来，接口方面的错误也已排除，软件测试的最后一步——确认测试即可开始。确认测试应检查软件能否按合同要求进行工作，即是否满足软件需求说明书中的确认标准。

实现软件确认要通过一系列黑盒测试。确认测试同样需要制订测试计划和过程，测试计划应规定测试的种类和测试进度，测试过程则定义一些特殊的测试用例，旨在说明软件与需求是否一致。无论是计划还是过程，都应该着重考虑软件是否满足合同规定的所有功能和性能，文档资料是否完整、准确，人机界面和其他方面（例如，可移植性、兼容性、错误恢复能力和可维护性等）是否令用户满意。

确认测试的结果有两种可能，一种是功能和性能指标满足软件需求说明的要求，用户可以接受；另一种是软件不满足软件需求说明的要求，用户无法接受。项目进行到这个阶段才发现严重错误和偏差一般很难在预定的工期内改正，因此必须与用户协商，寻求一个妥善解决问题的方法。

确认测试的另一个重要环节是配置复审。复审的目的在于保证软件配置齐全、分类有序，并且包括软件维护所必须的细节。

事实上，软件开发人员不可能完全预见用户实际使用程序的情况。例如，用户可能错误的理解命令，或提供一些奇怪的数据组合，亦可能对设计者自认明了的输出信息迷惑不解，等等。因此，软件是否真正满足最终用户的要求，应由用户进行一系列“验收测试”。验收测试既可以是非正式的测试，也可以有计划、有系统的测试。有时，验收测试长达数周甚至数月，不断暴露错误，导致开发延期。一个软件产品，可能拥有众多用户，不可能由每个用户验收，此时多采用称为α、β测试的过程，以便发现那些似乎只有最终用户才能发现的问题。

α测试是指软件开发公司组织内部人员模拟各类用户对即将面市的软件产品（称为α版本）进行测试，试图发现错误并修正。α测试的关键在于尽可能逼真地模拟实际运行环境和用户对软件产品的操作并尽最大努力涵盖所有可能的用户操作方式。经过α测试调整的软件产品称为β版本。紧随其后的β测试是指软件开发公司组织各方面的典型用户在日常工作中实际使用β版本，并要求用户报告异常情况、提出批评意见。然后软件开发公司再对β版本进行改错和完善。

a．验收测试过程

过程如下：

（a）软件需求分析：了解软件功能和性能要求、软硬件环境要求等，并特别要了解软件的质量要求和验收要求。

（b）编制《验收测试计划》和《项目验收准则》：根据软件需求和验收要求编制测试计划，制定需测试的测试项，制定测试策略及验收通过准则，并经过客户参与的计划评审。

（c）测试设计和测试用例设计：根据《验收测试计划》和《项目验收准则》编制测试用例，并经过评审。

（d）测试环境搭建：建立测试的硬件环境、软件环境等（可在委托客户提供的环境中进行测试）。

（e）测试实施：测试并记录测试结果。

（f）测试结果分析：根据验收通过准则分析测试结果，做出验收是否通过及测试评价。

（g）测试报告：根据测试结果编制缺陷报告和验收测试报告，并提交给客户。

b．验收测试的总体思路

用户验收测试是软件开发结束后，用户对软件产品投入实际应用以前进行的最后一次质量检验活动。它要回答开发的软件产品是否符合预期的各项要求，以及用户能否接受的问题。由于它不只是检验软件某个方面的质量，而是要进行全面的质量检验，并且要决定软件是否合格，因此验收测试是一项严格的正式测试活动。需要根据事先制定的计划，进行软件配置评审、功能测试、性能测试等多方面检测。

用户验收测试可以分为两个大的部分：软件配置审核和可执行程序测试，其大致顺序可分为：文档审核、源代码审核、配置脚本审核、测试程序或脚本审核、可执行程序测试。

要注意的是，在开发方将软件提交用户方进行验收测试之前，必须保证开发方本身已经对软件的各方面进行了足够的正式测试（当然，这里的“足够”，本身是很难准确定量的）。

用户在按照合同接收并清点开发方的提交物时（包括以前已经提交的），要查看开发方提供的各种审核报告和测试报告内容是否齐全，再加上平时对开发方工作情况的了解，基本可以初步判断开发方是否已经进行了足够的正式测试。

用户验收测试的每一个相对独立的部分，都应该有目标（本步骤的目的）、启动标准（着手本步骤必须满足的条件）、活动（构成本步骤的具体活动）、完成标准（完成本步骤要满足的条件）和度量（应该收集的产品与过程数据）。在实际验收测试过程中，收集度量数据，不是一件容易的事情。

（a）软件配置审核

对于一个外包的软件项目而言，软件承包方通常要提供如下相关的软件配置内容：

- 可执行程序、源程序、配置脚本、测试程序或脚本。
- 主要的开发类文档：《需求分析说明书》、《概要设计说明书》、《详细设计说明书》、《数据库设计说明书》、《测试计划》、《测试报告》、《程序维护手册》、《程序员开发手册》、《用户操作手册》、《项目总结报告》。
- 主要的管理类文档：《项目计划书》、《质量控制计划》、《配置管理计划》、《用户培训计划》、《质量总结报告》、《评审报告》、《会议记录》、《开发进度月报》。

在开发类文档中，容易被忽视的文档有《程序维护手册》和《程序员开发手册》。

《程序维护手册》的主要内容包括：系统说明（包括程序说明）、操作环境、维护过程、源代码清单等，编写目的是为将来的维护、修改和再次开发工作提供有用的技术信息。

《程序员开发手册》的主要内容包括：系统目标、开发环境使用说明、测试环境使用说明、编码规范及相应的流程等，实际上就是程序员的培训手册。

不同大小的项目，都必须具备上述的文档内容，只是可以根据实际情况进行重新组织。

对上述的提交物，最好在合同中规定阶段提交的时间，以免发生纠纷。

通常，正式的审核过程分为 5 个步骤：计划、预备会议（可选）、准备阶段、审核会议和问题追踪。预备会议是对审核内容进行介绍并讨论。准备阶段就是各责任人事先审核并记录发现的问题。审核会议是最终确定工作产品中包含的错误和缺陷。

审核要达到的基本目标是：根据共同制定的审核表，尽可能地发现被审核内容中存在的问题，并最终得到解决。在根据相应的审核表进行文档审核和源代码审核时，还要注意文档

与源代码的一致性。

在实际的验收测试执行过程中，常常会发现文档审核是最难的工作，一方面是由于市场需求等方面的压力使这项工作常常被弱化或推迟，造成持续时间变长，加大文档审核的难度；另一方面，文档审核中不易把握的地方非常多，每个项目都有一些特别的地方，而且也很难找到可用的参考资料。

（b）可执行程序的测试

在文档审核、源代码审核、配置脚本审核、测试程序或脚本审核都顺利完成后，就可以进行验收测试的最后一个步骤——可执行程序的测试，它包括功能、性能等方面的测试，每种测试也都包括目标、启动标准、活动、完成标准和度量等五部分。

要注意的是不能直接使用开发方提供的可执行程序用于测试，而要按照开发方提供的编译步骤，从源代码重新生成可执行程序。

在真正进行用户验收测试之前一般应该已经完成了以下工作（也可以根据实际情况有选择地采用或增加）：

- 软件开发已经完成，并全部解决了已知的软件缺陷。
- 验收测试计划已经过评审并批准，并且置于文档控制之下。
- 对软件需求说明书的审查已经完成。
- 对概要设计、详细设计的审查已经完成。
- 对所有关键模块的代码审查已经完成。
- 对单元、集成、系统测试计划和报告的审查已经完成。
- 所有的测试脚本已完成，并至少执行过一次，且通过评审。
- 使用配置管理工具且代码置于配置控制之下。
- 软件问题处理流程已经就绪。
- 已经制定、评审并批准验收测试完成标准。

具体的测试内容通常可以包括：安装（升级）、启动与关机、功能测试（正例、重要算法、边界、时序、反例、错误处理）、性能测试（正常的负载、容量变化）、压力测试（临界的负载、容量变化）、配置测试、平台测试、安全性测试、恢复测试（在出现掉电、硬件故障或切换、网络故障等情况时，系统是否能够正常运行）、可靠性测试等。

性能测试和压力测试一般情况下是在一起进行，通常还需要辅助工具的支持。在进行性能测试和压力测试时，测试范围必须限定在那些使用频度高的和时间要求苛刻的软件功能子集中。由于开发方已经事先进行过性能测试和压力测试，因此可以直接使用开发方的辅助工具。也可以通过购买或自己开发来获得辅助工具。具体的测试方法可以参考相关的软件工程书籍。

如果执行了所有的测试案例、测试程序或脚本，用户验收测试中发现的所有软件问题都已解决，而且所有的软件配置均已更新和审核，可以反映出软件在用户验收测试中所发生的变化，用户验收测试就完成了。

尽管测试阶段的划分十分明确，但是在具体的项目和产品的测试中，尤其在执行测试时，会根据实际需要来开展。关于软件测试阶段的理论可以参看朱少民写的《全程软件测试》一书中软件测试过程和开发过程的 V 模型关系。

测试的各个阶段所用时间比例是不同的，根据该阶段的性质，确定时间长短。当然，具体情况要具体掌握，根据不同项目不同情况，各测试阶段所用时间长短可随时调节。图 2-11

直观地描述出了测试各阶段所用的时间对比。

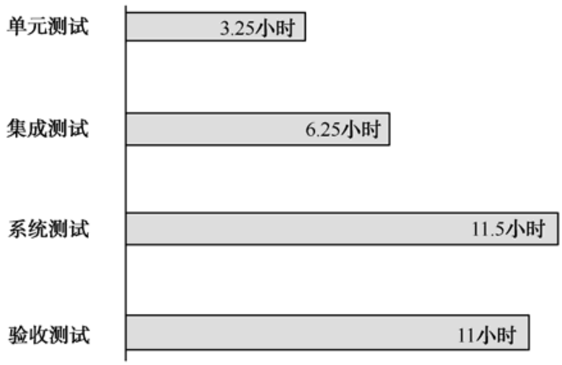


图 2-11　测试各阶段所用的时间

（2）软件测试的种类

对于测试种类的说法很多，最多的能达到几十种测试种类。但是实际工作中很多测试是互相包含的。按照企业中实际工作需要，通常主要进行下面几种类型的测试：功能测试、健壮性测试、接口测试、强度测试、压力测试、性能测试、用户界面测试、可靠性测试、安装/反安装测试、帮助文档测试。

下面介绍几种重要的测试种类及其测试的内容：

① 功能测试：功能测试主要针对产品需求说明书的测试，是验证功能是否适合需求，包括原定功能的检验、是否有冗余功能、遗漏功能。这类测试应由测试员做，这并不意味着程序员在发布前不必检查他们的代码能否工作，他们也需要进行基本功能的测试。

② 接口测试：程序员对各个模块进行系统联调的测试，包含程序内接口和程序外接口测试。这个测试，在单元测试阶段进行了一部分工作，而大部分都是在集成测试阶段完成的。由开发人员进行。

③ 性能测试：在交替进行负荷和强迫测试时常用的术语。性能测试关注的是系统的整体。它和通常所说的强度、压力/负载测试有密切关系。所以压力和强度测试应该与性能测试一同进行。

④ 用户界面测试：对系统的界面进行测试，测试用户界面是否友好、是否方便易用、设计是否合理、位置是否正确等一系列界面问题。

⑤ 安装/反安装测试：安装测试主要检验软件是否可以正确安装，安装文件的各项设置是否有效，安装后能否影响原系统；反安装是逆过程，测试是否删除干净，是否影响原系统等。

⑥ 文档测试：主要测试开发过程中针对用户的文档，以需求、用户手册、安装手册等为主，检验文档是否和实际应用存在差别。文档测试不需要编写测试用例。

测试种类的划分不要拘泥于上面的形式，总体来说应该服从于测试策略，可以根据具体工作的特点进行安排，为了工作更容易开展，完全可以把一些测试合并在一起进行。在后面的性能测试用例的编写上，充分体现了这一思想。

综合上面的分析，测试种类、测试阶段以及执行人员具体的关系如表 2-1 所示。

表 2-1　测试的种类、阶段和执行人员的关系

测 试 阶 段	测 试 类 型	执　行　者
单元测试	模块功能测试，包含部分接口测试、路径测试	开发工程师
集成测试	接口测试、路径测试，含部分功能测试	开发工程师（如果测试人员水平较高，可以由测试人员执行）
系统测试	功能测试、健壮性测试、性能测试、用户界面测试、安全性测试、压力测试、可靠性测试、安装/反安装测试	测试工程师
验收测试	对于实际项目来说基本同上，并包含文档测试；对于软件产品，主要测试相关的技术文档	测试工程师（根据实际需要，可能包含用户）

总之，测试的种类应该尽量的少，这样每次都可以执行更多的测试内容。例如在进行功能测试的同时，完全可以进行健壮性的测试（当然如果产品健壮性方面要求较高，就可以把健壮性测试作为独立的测试）。也就是说测试各个阶段中对执行不同的测试种类采用不同的测试技术。关于测试技术下面进行讲解。

3．黑盒测试中设计测试用例的基本方法

白盒测试和黑盒测试是软件测试中的两大方法，有时也将兼具两者特点的方法叫做灰盒测试。但传统的软件测试活动基本上都可以划到这两类方法当中。

黑盒测试注重于测试软件的功能性需求，也就是说黑盒测试要求软件工程师列出程序所有功能需求的输入条件。黑盒测试并不是白盒测试的替代品，而是用于辅助白盒测试发现其他类型的错误。黑盒测试主要用于测试的后期，一般由专门的测试人员来做。

本课程主要来学习黑盒测试，培养专门的测试人员。黑盒测试概括起来主要用来发现下面这些类型的错误：

- 功能错误或遗漏；
- 界面错误；
- 数据结构或外部数据库访问错误；
- 性能错误；
- 初始化和终止错误。

测试用例可以分为基本事件、备选事件和异常事件。设计基本事件的用例，应该参照用例规约（或设计规格说明书），根据关联的功能、操作按路径分析法设计测试用例。而对孤立的功能则直接按功能设计测试用例。基本事件的测试用例应包含所有需要实现的需求功能，覆盖率达 100%。

设计备选事件和异常事件的用例，则要复杂和困难得多。例如，字典的代码是唯一的，不允许重复。测试需要验证：字典新增程序中已存在有关字典代码的约束，若出现代码重复必须报错，并且报错文字正确。往往在设计编码阶段形成的文档对备选事件和异常事件分析描述不够详尽。而测试本身则要求验证全部非基本事件，并同时尽量发现其中的软件缺陷。

可以采用软件测试常用的基本方法：等价类划分法、边界值分析法、错误推测法、因果图法、逻辑覆盖法等设计测试用例。视软件的不同性质采用不同的方法。如何灵活运用各种基本方法来设计完整的测试用例，并最终实现暴露隐藏的缺陷，全凭测试设计人员的丰富经验和精心设计。

黑盒测试方法主要有五种，分为等价类划分法、边界值划分法、错误推测法、因果图法

和场景法。在实际测试用例设计过程中，不仅根据需要、场合单独使用这些方法，常常综合运用多个方法，使测试用例的设计更为有效。

这里先介绍边界值法，错误推测法和场景法。

（1）边界值法

边界值分析法是一种非常实用的测试用例设计技术，具有很强的发现程序错误的能力，它的测试用例来自于等价类的边界。大量测试工作的经验会告诉我们，大量的错误发生在输入或输出范围的边界上，而不是输入或输出范围的内部。边界值分析就是假定错误发生在输入或输出区间的边界上，因此使用边界值法设计测试用例，可以发现更多的错误。

在使用边界值法设计测试用例时，应该首先确定好输入边界和输出边界情况，然后选取正好等于、刚刚大于或刚刚小于边界的值作为测试数据，而不是选取等价类中的典型值或任意值作为测试数据。

一般情况下，可以遵循以下几个规则来设计测试用例：

① 如果输入条件规定了值的范围，应取刚达到这个范围的边界值，以及刚刚超过这个范围边界的值作为测试输入的数据。

② 如果输入条件规定了值的个数，应用最大个数、最小个数、比最小个数少一、比最大个数多一的数作为测试输入的数据。

③ 根据每个输入条件，使用规则一或二。

④ 如果程序的规格说明给出的输入域或输出域是有序集合，则应选取集合的第一个元素和最后一个元素作为测试用例数据。

⑤ 如果程序中使用了一个内部数据结构，应当选择这个内部数据结构的边界上的值来作为测试用例。

⑥ 分析规格说明，找出其他可能的边界条件。

下面举个例子让大家更深入地理解边界值法。

用户登录网上购物系统要购买某种商品，假设该商品剩余数量为 100 件，且用户只会输入整数。则用户只能购买 1～100 范围内的商品件数。使用边界值法设计测试用例，测试用户输入商品数量 Q 后，系统反应是否合乎标准。

提出边界时，一定要测试邻近边界的合法数据，即测试最后一个可能合法的数据，以及刚刚超过边界的非法数据。越界测试通常简单地加 1 或者用最小的数减 1。图 2-12 中，输入分区将 1～100 分成三个区间，再考虑上恰好等区间边界的情况，我们可以考虑商品数量 Q 的输入区间如下：

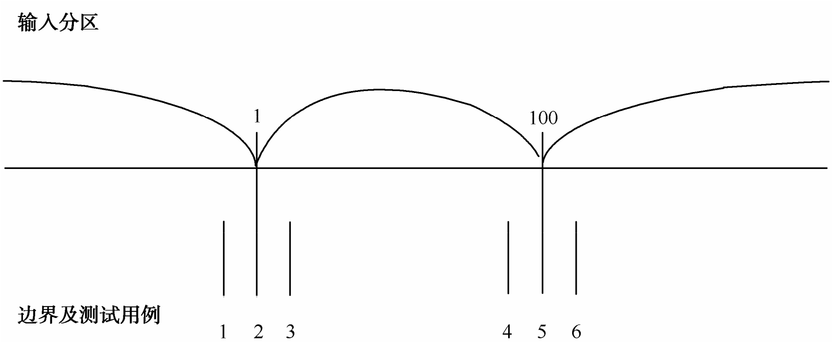


图 2-12　边界值分析

① $Q<1$

② $Q=1$

③ $1<Q<100$

④ $Q=100$

⑤ $Q>100$

根据上面的分析可以设计六个用例：

① Test Case 1：输入 0，返回错误信息“您必须输入大于等于一个数量值”。

② Test Case 2：输入 1，页面正确运行。

③ Test Case 3：输入 2，页面正确运行。

④ Test Case 4：输入 99，页面正确运行。

⑤ Test Case 5：输入 100，页面正确运行。

⑥ Test Case 6：输入 101，返回错误信息“您所选购的商品数量仅剩 100 件”。

测试员可以将上面的信息填入用例设计表格中，形成标准的测试用例。

（2）错误推测法

错误推测法就是根据经验和直觉推测程序中所有可能存在的各种错误，从而有针对性地设计测试用例的方法。

使用错误推测法时，可以凭经验列举出程序中所有可能有的错误和容易发生错误的特殊情况，帮助猜测错误可能发生的位置，提高错误猜测的有效性，根据他们选择测试用例。

例如：输入表格为空格；输入数据和输出数据为 0 的情况。

（3）场景法

场景是通过描述经过用例的路径来确定的过程，这个过程要从用例开始到结束遍历其中所有基本流和备选流。场景法就是根据这些基本流和备选流的流动过程来设计测试用例。

目前的软件几乎都是由事件触发来控制流程的，事件触发时的情景便形成了场景，而同一事件不同的触发顺序和处理结果形成事件流。这种在软件设计方面的思想也可被引入到软件测试中，生动地描绘出事件触发时的情景，有利于测试设计者设计测试用例，同时测试用例也更容易的得到理解和执行。提出这种测试思想的是 Rational 公司。

下面使用网上购物系统的购物场景举例说明。

① 场景描述

用户进入网上购物系统网站进行购物，选好物品后进行购买，这时需要使用账号登录，登录成功后付款，交易成功后生成订单，完成此次购物活动。

② 使用场景法设计测试用例

a．确定基本流和备选流事件

表 2-2 列出了用户购物活动中的基本流和备选流。

表 2-2　基本流和备选流

基　本　流	登录网上购物系统网站，选择物品，登录账号，付钱交易，生成订单
备选流 1	账号不存在
备选流 2	账号或密码错误
备选流 3	用户账号余额不足
备选流 4	用户账号没有钱
备选流 5	用户退出系统

b．根据基本流和备选流来确定场景

表 2-3 列出了每个场景中包括的基本流和备选流，需要特别说明的是，由于场景 1 中用户成功购买了物品，所以只有基本流而不需经过备选流。

表 2-3　场景

场景 1：成功购物	基本流	
场景 2：账号不存在	基本流	备选流 1
场景 3：账号或密码错误	基本流	备选流 2
场景 4：用户账号余额不足	基本流	备选流 3
场景 5：用户账号没有钱	基本流	备选流 4

c．设计用例

对每一个场景都要做测试用例，可以使用矩阵（表格）来管理用例。用行表示各个测试用例，列表示测试用例的信息。首先将测试用例的 ID、条件、涉及的数据元素以及预期结果列在矩阵中，然后将这些数据确定下来，填写在表格中。

在表 2-4 中，“有效”表示这个条件必须是有效的才可执行基本流，而“无效”用于表示这种条件下将激活所需备选流。“不适用”表示这个条件不适用于测试用例。

表 2-4　测试用例信息

测试用例 ID	场景/条件	账　号	密　码	用户账号余额	预 期 结 果
1	场景 1：成功购物	有效	有效	有效	成功购物
2	场景 2：账号不存在	无效	不适用	不适用	提示账号不存在
3	场景 3：账号或密码错误（账号正确，密码错误）	有效	无效	不适用	提示账号或密码错误，返回基本流步骤 3
4	场景 3：账号或密码错误（账号错误，密码正确）	无效	有效	不适用	提示账号或密码错误，返回基本流步骤 3
5	场景 4：用户账号余额不足	有效	有效	无效	提示账号余额不足请充值
6	场景 5：用户账号没有钱	有效	有效	无效	提示账号余额请充值

d．设计数据，填入表 2-4，结果如表 2-5 所示

表 2-5　测试用例数据

测试用例 ID	场景/条件	账号	密码	用户账号余额	预期结果
1	场景 1：成功购物	wangsh	Passw0rd	193	成功购物，用户账号余额正确
2	场景 2：账号不存在	song	不适用	不适用	提示账号不存在
3	场景 3：账号或密码错误（账号正确，密码错误）	wangsh	666666	不适用	提示账号或密码错误，返回基本流步骤 3
4	场景 3：账号或密码错误（账号错误，密码正确）	song	passw0rd	不适用	提示账号或密码错误，返回基本流步骤 3
5	场景 4：用户账号余额不足	wsh	pass0rd	2	提示账号余额不足请充值
6	场景 5：用户账号没有钱	sunxx	817217	0	提示账号余额请充值

这样我们就完成了整个测试用例的设计过程。

2.1.6　拓展任务

独立编写天天超市管理系统用户管理模块的测试用例集。

工作任务 2.2　Test Suite商品管理

2.2.1　Test Suite商品类别管理

1．工作任务描述

管理员成功登录系统后，进入图 2-13 所示的商品类别浏览界面，单击相应类别的修改或删除按钮进行商品类别的管理。其中商品类别添加界面如图 2-14 所示，商品类别修改界面如图 2-15 所示。

本节任务是编写商品类别管理功能的测试用例集，分别设计浏览商品类别，添加商品类别和修改商品类别的测试用例。设计测试用例的基本方法为场景法、边界值法和错误推测法。

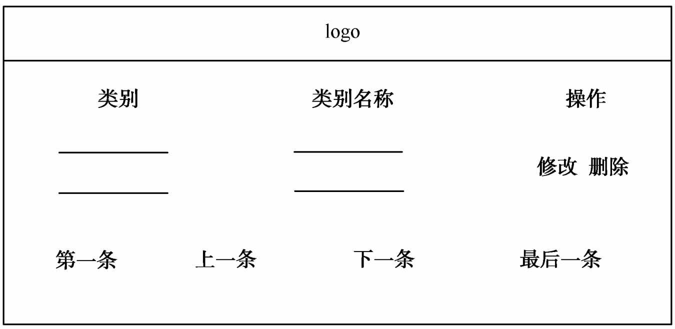


图 2-13　商品类别浏览界面

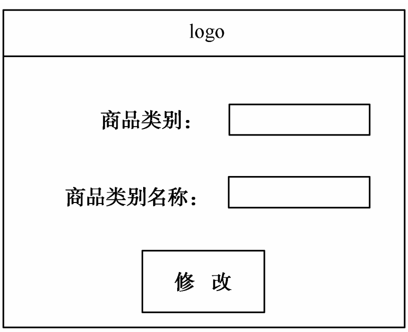


图 2-14　商品类别添加界面

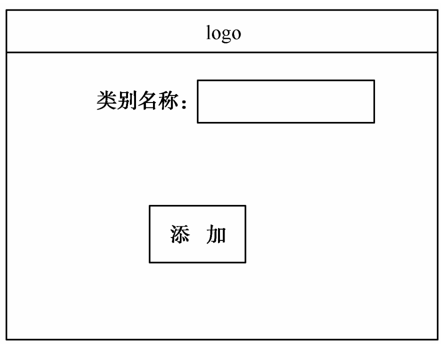


图 2-15　商品类别修改界面

工作过程：

（1）编写商品类别添加的测试用例集

Test Case 047：必填项是否允许为空
Summary： 检验系统是否对必填项为空的情况做了处理

续表

Steps： 1．单击［商品类别］\［添加］按钮 2．什么都不输入，直接单击［添加］按钮	Expected Results： 1．弹出“商品类别添加界面” 2．提示“类别名称不能为空”
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 048：输入字符数等于域允许的最大字符数	
Summary： 系统是否对域的输入长度进行了检验	
Steps： 1．单击［商品类别］\［添加］按钮 2．在“类别名称”中输入“国产电视机”，单击［添加］按钮	Expected Results： 1．弹出“商品类别添加界面” 2．提示“保存成功”
边界值法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 049：输入字符数大于域允许的最大字符数	
Summary： 检验系统是否对域输入长度进行了验证	
Steps： 1．单击［商品类别］\［添加］按钮 2．在“类别名称”中输入“国产电脑主机配件”，单击［添加］按钮	Expected Results： 1．弹出“商品类别添加界面”； 2．提示“您输入的字符数过多，请限制在 5 个汉字。”
边界值法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 050：回车验证	
Summary： 检验系统是否对回车键进行了处理	
Steps： 按［Enter］键	Expected Results： 相当于单击了［添加］按钮
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 051：验证系统定义的域长度是否够用	
Summary： 输入有现实意义且字符数比较多的类别名称，检验系统是否允许输入	
Steps： 1．单击［商品类别］\［添加］按钮 2．在“类别名称”中输入“国产电子元器件”，单击［添加］按钮	Expected Results： 1．弹出“商品类别添加界面”； 2．提示“保持成功”
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 052：重名校验	
Summary： 检验系统是否对类别名重名的情况做了校验	
Steps： 1．单击［商品类别］\［添加］按钮 2．在“类别名称”中输入“国产电视机”，单击［添加］按钮	Expected Results： 1．弹出“商品类别添加界面”； 2．提示“类别名称已存在”
场景法	
Pass/Fail：	Test Notes：
Author admin	

（2）编写类别修改的测试用例集

Test Case 053：必添项是否允许为空	
Summary： 检验系统是否对必添项为空的情况做了处理。	
Steps： 1．单击［商品类别］\［浏览］按钮 2．单击商品类别“国产电视机”后面的链接［修改］ 3．清空类别名称，单击［修改］按钮	Expected Results： 1．弹出“商品类别浏览界面” 2．弹出“商品类别修改界面” 3．提示“商品类别名称不能为空”
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 054：输入字符数等于域允许的最大字符数	
Summary： 系统是否对域的输入长度进行了检验	
Steps： 1．单击［商品类别］\［浏览］按钮 2．单击商品类别“国产电视机”后面的链接［修改］ 3．将类别名称改为“五金零配件”，单击［修改］按钮	Expected Results： 1．弹出“商品类别浏览界面” 2．弹出“商品类别修改界面” 3．提示“修改成功”
边界值法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 055：输入字符数大于域允许的最大字符数	
Summary： 检验系统是否对域输入长度进行了验证	
Steps： 1．单击［商品类别］\［浏览］按钮 2．单击商品类别“国产电视机”后面的链接［修改］ 3．将类别名称改为“国产电子元器件”，单击［修改］按钮	Expected Results： 1．弹出“商品类别浏览界面” 2．弹出“商品类别修改界面” 3．提示“您输入的名称过长，请重新输入”
边界值法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 056：不修改直接保存	
Summary： 不修改，直接单击［保存］按钮	
Steps： 1．单击［商品类别］\［浏览］按钮 2．单击商品类别“国产电视机”后面的链接［修改］ 3．不修改直接单击［修改］按钮	Expected Results： 1．弹出“商品类别浏览界面” 2．弹出“商品类别修改界面” 3．提示“修改成功”，不应提示类别名称已存在
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 057：回车验证	
Summary： 检验系统是否对回车键进行了处理	
Steps： 按［Enter］键	Expected Results： 相当于单击了［修改］按钮
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 058：重名校验	
Summary： 检验系统是否对类别名重名的情况做了校验	
Steps： 1．单击［商品类别］\［添加］按钮 2．在“类别名称”中输入“长虹电视”，单击［添加］按钮 3．单击［商品类别］\［浏览］按钮 4．单击商品类别“国产电视机”后面的链接［修改］ 5．将类别名称改为“长虹电视”，单击［修改］按钮	Expected Results： 1．弹出“商品类别添加界面” 2．提示“添加成功” 3．弹出“商品类别浏览界面” 4．弹出“商品类别修改界面” 5．提示“类别名称已存在”
场景法	
Pass/Fail：	Test Notes：
Author admin	

（3）编写类别删除的测试用例集

Test Case 059：删除未被使用的类别	
Summary： 未被使用的类别可以被删除	
Steps： 1．单击［商品类别］\［浏览］按钮 2．选择未被使用的类别，单击类别名称后面的链接文字［删除］	Expected Results： 1．打开“商品类别浏览界面” 2．提示“删除成功”
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 060：删除已被使用的类别	
Summary： 已被使用的类别不可以被删除	
Steps： 1．在添加商品类别“电脑”； 2．添加商品“hp 家用电脑”，商品类别选择“电脑” 3．单击［商品类别］\［浏览］按钮 4．选择已被使用的类别“电脑”，单击类别名称后面的链接文字［删除］	Expected Results： 1．商品类别添加成功 2．商品添加成功 3．打开商品类别浏览界面 4．提示“该商品类别已被使用不能删除”
场景法	
Pass/Fail：	Test Notes：
Author admin	

（4）编写类别浏览、查看的测试用例集

Test Case 061：查看信息显示是否完整	
Summary： 所有已添加的类别均应显示出来	
Steps： 单击［查看］（［浏览］）按钮，检查类别显示是否完整、类别内容是否正确	Expected Results： 所有添加的类别均可以显示出来；显示的类别名称与添加时所填写的内容一致
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 062：翻页	
Summary： 单击翻页按钮可以正确跳转到相应的页面上	
Steps： 1．单击类别［查看］（［浏览］）按钮 2．单击［第一条］、［下一条］、［上一条］、［最后一条］	Expected Results： 1．弹出商品类别浏览界面 2. 可以正确跳转到对应的页面上，不存在部分信息重复显示，部分信息显示不出来的情况
场景法	
Pass/Fail：	Test Notes：
Author admin	

2.2.2 Test Suite商品管理

1．工作任务描述

网上购物网站必然包含大量的商品信息，管理员不仅要管理商品的类别，还要对商品本身进行管理，需要添加和修改商品信息。商品管理模块可以为商品设定不同的属性，如商品的名称、规格、售价、生产厂商及商品的图片等，可以方便地编辑丰富的商品信息呈现方式，及时调整商品信息。商品信息添加的界面如图 2-16 所示，商品信息修改的界面如图 2-17 所示。

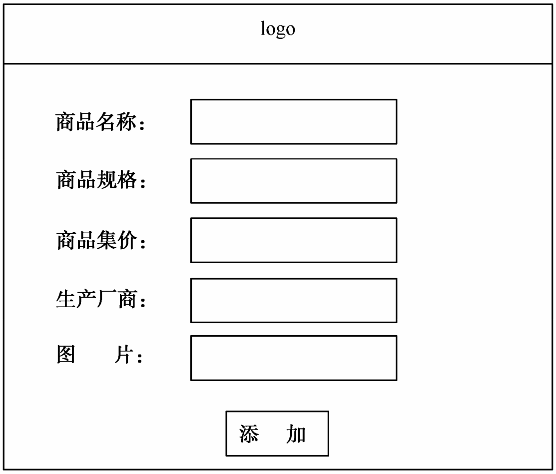


图 2-16　商品信息添加界面

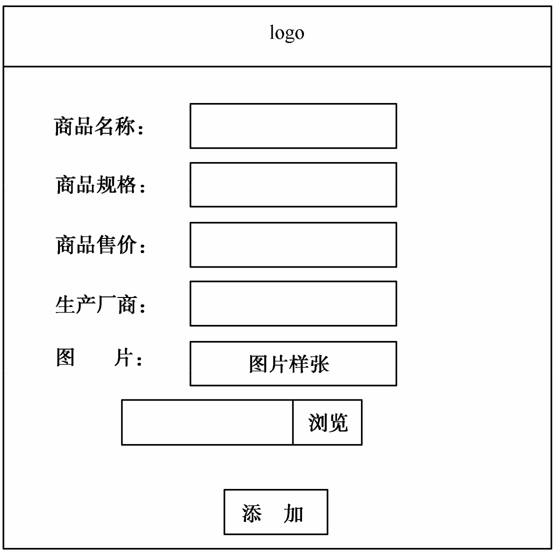


图 2-17　商品信息修改界面

2．工作过程

（1）编写商品添加的测试用例集

Test Case 063：必填项是否允许为空
Summary： 检验系统是否对必填项为空的情况做了处理

续表

<table>
<tr><td>Steps：
1．单击［商品］\［添加］按钮
2．什么都不输入，直接单击［添加］按钮；</td><td>Expected Results：
1．弹出“商品添加界面”
2．提示“商品名称、商品类别、商品规格、商品售价、生产商、图片不能为空”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 064：输入字符数等于域允许的最大字符数</td></tr>
<tr><td colspan="2">Summary：
系统是否对域的输入长度进行了检验</td></tr>
<tr><td>Steps：
1．单击［商品］\［添加］按钮
2．在添加界面中输入以下内容
商品名称：一二三四五六七八九十
商品类别：一二三四五
商品规格：一二三四五六
商品售价：9999.99
生产商：一二三四五六七八九十一二三四五六七八九十一二三四五
上传图片，图片名：一二三四五六七八.bmp
单击［添加］按钮</td><td>Expected Results：
1．弹出“商品添加界面”；
2．提示“保存成功”</td></tr>
<tr><td colspan="2">边界值法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 065：输入字符数大于域允许的最大字符数</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对域输入长度进行了验证</td></tr>
<tr><td>Steps：
1．单击［商品］\［添加］按钮
2．在添加界面中输入以下内容
商品名称：一二三四五六七八九十一
商品类别：一二三四五一
商品规格：一二三四五六一
商品售价：99999.999
生产商：一二三四五六七八九十一二三四五六七八九十一二三四五一
上传图片，图片名：一二三四五六七八一.bmp
单击［添加］按钮</td><td>Expected Results：
1．弹出“商品添加界面”；
2．提示“您输入的‘商品名称、商品类别、商品规格、商品售价、生产商、上传图片’字符数过多，请重新输入。”</td></tr>
<tr><td colspan="2">边界值法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 066：回车验证</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对回车键进行了处理</td></tr>
<tr><td>Steps：
按［Enter］键</td><td>Expected Results：
相当于单击了［添加］按钮。</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 067：验证系统定义的域长度是否够用</td></tr>
<tr><td colspan="2">Summary：
输入有现实意义且字符数比较多的商品信息，检验系统是否允许输入</td></tr>
<tr><td>Steps：
1．单击［商品类别］\［添加］按钮
2．在添加界面中输入以下内容
商品名称：hp 笔记本电脑
商品类别：电脑
商品规格：1 台
商品售价：19999.99
生产商：惠普中国
上传图片，图片名：hp_computer.bmp
单击［添加］按钮</td><td>Expected Results：
1．弹出“商品添加界面”；
2．提示“保持成功”</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 068：重名校验</td></tr>
<tr><td colspan="2">Summary：
检验系统是否对重名的情况做了校验</td></tr>
<tr><td>Steps：
1．单击［商品］\［添加］按钮
2．在“商品名称”中输入“长虹电视机 2501”，单击［添加］按钮
3．再次单击［商品］\［添加］按钮
4．在“商品名称”中输入“长虹电视机 2501”，单击［添加］按钮</td><td>Expected Results：
1．弹出“商品添加界面”；
2．提示“添加成功”
3．弹出“商品添加界面”；
4．提示“商品已存在”</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

（2）编写商品修改的测试用例集

Test Case 069：必添项是否允许为空	
Summary： 检验系统是否对必填项为空的情况做了处理	
Steps： 1．单击［商品］\［浏览］按钮 2．单击商品“长虹电视机 2501”后面的链接［修改］ 3．清空所有输入域，单击［修改］按钮	Expected Results： 1．弹出“商品浏览界面” 2．弹出“商品修改界面” 3．提示“商品名称、商品类别、商品规格、商品售价、生产商、图片不能为空”
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 070：输入字符数等于域允许的最大字符数	
Summary： 系统是否对域的输入长度进行了检验	
Steps： 1．单击［商品］\［浏览］按钮 2．单击商品“长虹电视机 2501”后面的链接［修改］ 3．将商品信息修改为以下内容 商品名称：一二三四五六七八九一 商品类别：一二三四一 商品规格：一二三四五一 商品售价：9999.91 生产商：一二三四五六七八九十一二三四五六七八九十一二三四一 上传图片，图片名：一二三四五六七一.bmp 单击［修改］按钮	Expected Results： 1．弹出“商品浏览界面” 2．弹出“商品修改界面” 3．提示“修改成功”
边界值法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 071：输入字符数大于域允许的最大字符数	
Summary： 检验系统是否对域输入长度进行了验证	
Steps： 1．单击［商品］\［浏览］按钮 2．单击商品“一二三四五六七八九一”后面的链接［修改］ 3．将商品信息修改为以下内容 商品名称：一二三四五六七八九十一 商品类别：一二三四五一 商品规格：一二三四五六一 商品售价：99999.999	Expected Results： 1．弹出“商品浏览界面” 2．弹出“商品修改界面” 3．提示“您输入的‘商品名称、商品类别、商品规格、商品售价、生产商、图片’过长，请重新输入”

生产商：一二三四五六七八九十一二三四五六七八九十一二三四五一 上传图片，图片名：一二三四五六七八九十一.bmp 单击［修改］按钮	
边界值法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 072：不修改直接保存	
Summary： 不修改，直接单击［保存］按钮	
Steps： 1．单击［商品］\［浏览］按钮 2．单击商品“长虹电视机 2501”后面的链接［修改］ 3．不修改，直接单击［修改］按钮	Expected Results： 1．弹出“商品浏览界面” 2．弹出“商品修改界面” 3．提示“修改成功”，不应提示“商品名称已存在”
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 073：回车验证	
Summary： 检验系统是否对回车键进行了处理	
Steps： 按［Enter］键	Expected Results： 相当于单击了［修改］按钮
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 074：重名校验	
Summary： 检验系统是否对重名做了校验	
Steps： 1．单击［商品］\［添加］按钮 2．添加一个商品，商品名称：“儿童电动车“，单击［添加］按钮 3．单击［商品］\［浏览］按钮 4．单击商品“长虹电视机 2501”后面的链接［修改］ 5．将商品名称改为“儿童电动车”，单击［修改］按钮；	Expected Results： 1．弹出“商品添加界面” 2．提示“添加成功” 3．弹出“商品浏览界面” 4． 弹出“商品修改界面” 5．提示“商品名称已存在”
场景法	
Pass/Fail：	Test Notes：
Author admin	

（3）编写商品删除的测试用例集

Test Case 075：删除未被使用的商品	
Summary： 未被使用的商品可以被删除	
Steps： 1．单击［商品］\［浏览］按钮 2．选择未被使用的商品，单击商品名称后面的链接文字［删除］	Expected Results： 1．打开商品浏览界面 2．提示“删除成功”
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 076：删除已被使用的类别	
Summary： 已被使用的商品不可以被删除	
Steps： 1．在添加商品“诺基亚 N7300” 2．注册用户“米奇”查看“诺基亚 N7300”的商品详细信息，单击［购买］按钮 3．单击［商品］\［浏览］按钮 4．选择已被使用的商品“诺基亚 N7300”，单击商品名称后面的链接文字［删除］	Expected Results： 1．商品添加成功 2．将“诺基亚 N7300”添加到自己的购物车中 3．打开商品浏览界面 4．提示“该商品已被使用不能删除”
场景法	
Pass/Fail：	Test Notes：
Author admin	

（4）编写商品浏览的测试用例集

Test Case 077：查看信息显示是否完整	
Summary： 所有已添加的商品均应显示出来	
Steps： 单击［浏览］按钮，检查商品显示是否完整、商品内容是否正确	Expected Results： 所有添加的商品均可以显示出来；显示的商品信息与添加时所填写的内容一致
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 078：翻页
Summary： 单击翻页按钮可以正确跳转到相应的页面上

Steps： 1．单击商品［浏览］按钮 2．单击［第一条］、［下一条］、［上一条］、［最后一条］	Expected Results： 1．弹出商品浏览界面 2．可以正确跳转到对应的页面上，不存在部分信息重复显示，部分信息显示不出来的情况
场景法	
Pass/Fail：	Test Notes：
Author admin	

2.2.3 应知应会

1．黑盒测试中测试用例设计的其他方法

在第一节中，我们详细介绍了边界值分析法、错误推测法和场景法，这里我们介绍等价类划分法和因果图法。

（1）等价类划分法

等价类划分法是黑盒测试的典型方法，只需按照需求文档中对系统的要求和说明对输入的范围进行划分，然后从每个区域内选取一个有代表性的测试数据，完全不用考虑系统的内部结构。如果等价类划分得合理，选取的这个数据就代表了这个区域内所有的数据。

具体来讲，等价类划分法就是把所有可能的输入数据，即程序的输入域划分成若干部分（子集），然后从每一个子集中选取少数具有代表性的数据作为测试用例。其中每个输入域的集合（子集）就是等价类，在这个集合中每个输入条件都是等效的，如果其中一个的输入不导致问题发生，那么这个等价类中其他输入也不会发生错误。

等价类分为有效等价类和无效等价类。有效等价类就是由那些对程序的规格说明有意义的、合理的输入数据所构成的集合，利用有效等价类可检验程序是否实现了需求文档中所规定的功能和性能。无效等价类就是那些对程序的规格说明不合理的或无意义的输入数据所构成的集合。

划分等价类最重要的是集合的划分。集合要划分为互不相交的子集，而子集的并是整个集合。确定等价类的原则如下：

① 在输入条件规定了取值范围（闭区间）或值的个数的情况下，则可以确定一个有效等价类和两个无效等价类。

② 在输入条件规定了输入值的集合或者规定了“必须如何”的条件的情况下，可确定一个有效等价类和一个无效等价类。

③ 在输入条件是一个布尔量的情况下，可确定一个有效等价类。

④ 在规定了输入数据的一组值（假定 n 个），并且程序要对每一个输入值分别处理的情况下，可确定 n 个有效等价类和一个无效等价类。

⑤ 在规定了输入数据必须遵守的规则的情况下，可确定一个有效等价类（符合规则）和若干个无效等价类（从不同角度违反规则）。

⑥ 在确知已划分的等价类中各元素在程序处理中的方式不同的情况下，则应再将该等价类进一步的划分为更小的等价类。

在这里我们还使用前面介绍边界值法时的例子，来说明等价类划分法如何使用。

前面我们假设用户购买某种商品时只剩余 100 件，并且用户只会输入整数 Q。那么在这个例子中我们如何划分等价类呢？根据输入要求，将输入区域划分为 3 个等价类，如图 2-18 所示。

输入域分成了一个有效等价类（1 到 100 之间）和两个无效等价类（小于 1 和大于 100），将这些等价类填入下表中，如表 2-6 所示。

表 2-6　等价类

测试用例 ID	所属等价类	用户输入数量	预 期 结 果
1	1	−9	提示“请输入 1～100 之间的整数”
2	2	87	成功购物
3	3	189	提示“请输入 1～100 之间的整数”

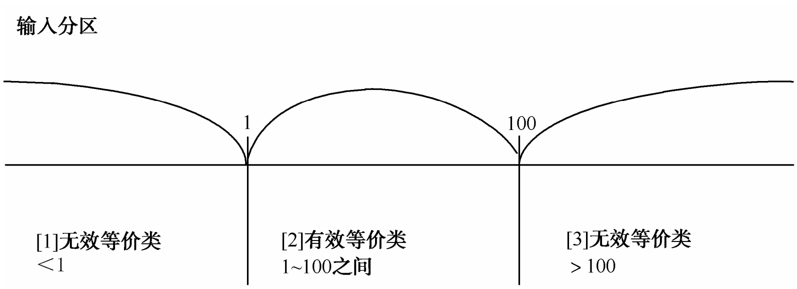


图 2-18　等价类划分法

当然，上面我们只是假设用户只会输入整数，但是用户输入小数、字母、汉字甚至是其他符号的情况是肯定存在的，这说明我们的等价类很不完善，只考虑了输入数据的范围却没考虑输入数据的类型，那么综合考虑所有情况应该如何设计等价类呢？这个问题留给读者去思考。

通过上面的例子我们可以想到，边界值法和等价类划分法是有紧密联系的。边界值法是对等价类划分法的补充，边界值其实就是在划分等价类的过程中产生的，正如前面边界值法中所述，正是由于等价类划分的区域边界的地方最容易出错，在从等价类中选取测试数据的时候也经常选取边界值。

（2）因果图法

因果图法是一种适合于描述对于多种条件的组合相应产生多个动作的形式的测试用例设计方法。“因”指的就是输入，“果”指的就是输出。因果图法也是一种黑盒测试技术，但没有边界值法、等价类法和错误推测法常用。

使用因果图法的基本步骤如下：

① 分析软件规格说明描述中哪些是原因，哪些是结果，并给每个原因和结果赋予一个标识符。

② 分析软件规格说明描述的语义。找出原因和结果之间、原因和原因之间的关系，根据这些关系，画出因果图。

③ 在因果图上用一些记号表明约束或限制条件。

④ 把因果图转换为判定表。

⑤ 把判定表的每一列拿出来作为依据，设计测试用例。

以上是黑盒测试中测试用例设计的基本方法，这些方法不是单独存在的，每个测试项目都会用到多种方法，针对不同类型的软件使用不同的测试方法非常重要。在实际测试中，经常需要综合使用各种方法才能有效地提高测试效率和测试覆盖度，这就需要认真地研究这些方法，积累更多的测试经验，提高测试水平。

2．测试用例编写模板

编写测试用例文档应有文档模板，须符合内部的规范要求。测试用例文档将受制于测试用例管理软件的约束。统一测试用例编写的规范，为测试设计人员提供测试用例编写的指导，提高编写的测试用例的可读性，可执行性、合理性。为测试执行人员更好执行测试，提高测试效率，最终提高公司整个产品的质量。

软件产品或软件开发项目的测试用例一般以该产品的软件模块或子系统为单位，形成一个测试用例文档，但并不是绝对的。

测试用例文档由简介和测试用例两部分组成。简介部分编制了测试目的、测试范围、定义术语、参考文档、概述等。测试用例部分逐一列示各测试用例。每个具体测试用例都将包括下列详细信息：用例编号、用例名称、测试等级、入口准则、验证步骤、期望结果（含判断标准）、出口准则、注释等。以上内容涵盖了测试用例的基本元素：测试索引，测试环境，测试输入，测试操作，预期结果，评价标准。

在实际应用中，很多称之为模板与规范的东西并不一定严格定义，而应该根据具体情况运用适合的模板与规范，这样在实践中才能灵活运用，而不流于形式和教条化。表 2-7 是测试用例的一种写法。

表 2-7　测试用例设计模板之一

测试用例 ID	测试用例的 ID（由案例管理系统自动生成，方便跟踪管理）		
测试用例名称			
产品名称		产品版本	
功能模块名		测试平台	
用例入库者		用例更新者	
用例入库时间		用例更新时间	
测试功能点	测试的功能检查点		
测试目的	该测试案例的测试目的		
测试级别	测试级别：主路径测试、烟雾测试、基本功能测试、详细功能测试		
测试类型	测试类型：功能测试、边界测试、异常测试、性能测试、压力测试、兼容测试、安全测试、恢复测试、安装测试、界面测试、启动/停止测试、文档测试、配置测试、可靠性测试、易用性测试、多语言测试		
预置条件	对测试的特殊条件或配置进行说明		
测试步骤	详细描述测试过程，案例的操作步骤建议少于 15 个		
预期结果	预期的测试结果		

在本书所写的测试用例设计与表 2-7 所示模板有所不同，当然也是在实际使用过程中对它进行了一些删减与变化。下面再给出几种测试用例设计模板，如表 2-8、表 2-9 所示，读者可以根据需要取舍。

表 2-8　测试用例设计模板之二

用 例 编 号	
测试优先级	
用例摘要	
测试类型	
用例类型	
用例设计者	
设计日期	
对应需求编号	
对应 UI	
对应 UC	
版本号	
对应开发人员	
前置条件	
测试方法	
输入数据	
执行步骤	（需详细写出执行步骤）
预期输出	
实际结果	
测试日期	
结论	

表 2-9　测试用例设计模板之三

测试用例				测试记录	
用例编号	测试目的/对应需求	输入\预置条件	预期输出	操作过程	结果
					功能正确
					功能不正确

续表

2.2.4　拓展任务

独立完成天天超市用例编写任务。

工作任务 2.3　Test Suite购物管理

2.3.1　Test Suite商品查看

1．工作任务描述

客户成功登录系统后，可以进行网上购物，选择商品加入购物车。如果需要查看自己所选购商品，则可以进入如图 2-19 所示的页面，单击上一条、下一条按钮滚动翻看。在这个页面中，客户可以单击查询按钮来查询自己所需要的商品，并且可以单击查看购物车看到自己已经选购的商品。

本节任务就是编写商品查看功能的测试用例集。

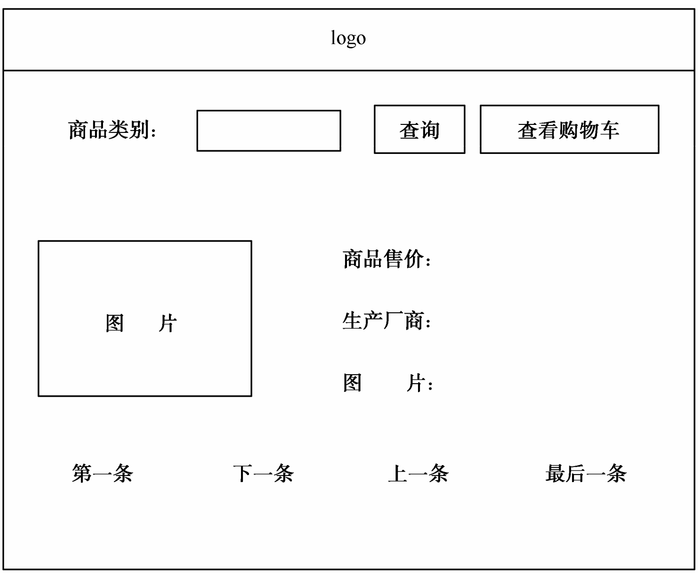


图 2-19　商品查看界面

2．工作过程

（1）编写商品查看的测试用例集

<table>
<tr><td colspan="2">Test Case 079：翻页</td></tr>
<tr><td colspan="2">Summary：
单击翻页按钮可以正确跳转到相应的页面上</td></tr>
<tr><td>Steps：
1．“米奇”登录
2．在页面中单击［第一条］、［下一条］、［上一条］、［最后一条］</td><td>Expected Results：
1．弹出“米奇”的个人购物主页
2．可以正确跳转到对应的页面上，不存在部分信息重复显示，部分信息显示不出来的情况</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass / Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

（2）编写商品查询的测试用例集

<table>
<tr><td colspan="2">Test Case 080：不输入查询</td></tr>
<tr><td colspan="2">Summary：
不输入信息，直接单击［查询］按钮</td></tr>
<tr><td>Steps：
不输入查询条件，直接单击［查询］按钮</td><td>Expected Results：
可以查询到全部商品</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 081：选择特定类别查询</td></tr>
<tr><td colspan="2">Summary：
在类别下拉列表中选择一个类别，执行查询</td></tr>
<tr><td>Steps：
在类别下拉列表中选择一个特定类别，单击［查询］按钮</td><td>Expected Results：
可以查询到该类别下的全部商品</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 082：模糊查询</td></tr>
<tr><td colspan="2">Summary：
输入类别的部分信息，执行查询</td></tr>
<tr><td>Steps：
在查询条件输入框中输入不完整的类别信息，如查询“九阳豆浆机”，只输入“九阳”，单击［查询］按钮</td><td>Expected Results：
可以查询到所有商品类别中包含“九阳”字样的商品</td></tr>
<tr><td colspan="2">场景法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

Test Case 083：在查询条件中输入特殊字符	
Summary： 系统应过滤或不允许输入特殊字符	
Steps： 在查询条件中输入“九阳”	Expected Results： 预期一：系统自动将特殊字符过滤掉，并可以查询到所有“九阳”相关的产品 预期二：系统提示“查询条件中不允许包含特殊字符‘’，请重新输入”
场景法	
Pass/Fail：	Test Notes：
Author admin	

2.3.2　Test Suite购买商品

1．工作任务描述

购物车是一个仿照显示商场中的人性化的工具，浏览者对于中意的商品在购买前临时存放在购物车中，并可以随时增减购物车中的商品种类和数量，以提高购物效率。购物车可以对注册及非注册用户使用以简化购物流程从而激起用户潜在购买欲望。

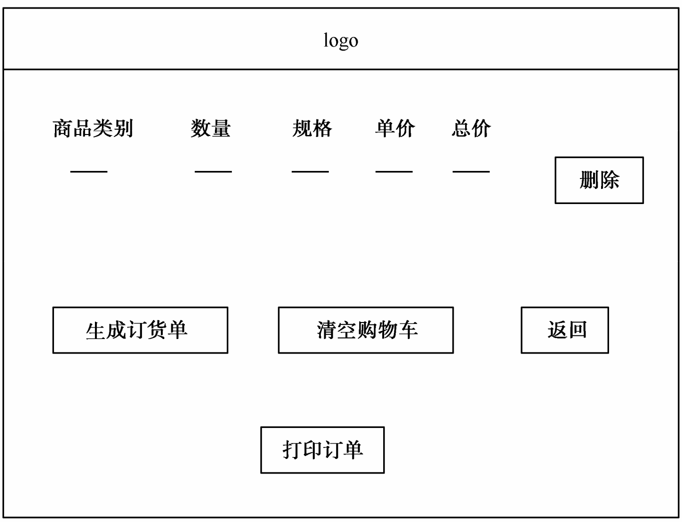


图 2-20　购买商品界面

网上购物系统中也使用了购物车工具。客户选择完商品后，在如图 2-19 所示的页面上点查看购物车，可以进入如图 2-20 所示的购买页面，在此有所选商品的详细列表信息，对不满意的商品可以选择删除。待完全确定所购物品后，客户可以单击生成订货单购买物品。

本节任务就是编写购买商品和购物车管理的测试用例集。

2．工作过程

（1）编写购买商品的测试用例集

Test Case 084：购买商品	
Summary： 单击［购买］按钮，可以将商品放入购物车	
Steps： 注册用户“米奇”查看“诺基亚 N7300”的商品详细信息，输入购买数量“1”，单击［购买］按钮	Expected Results： 商品“诺基亚 N7300”添加到“米奇”的购物车中
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 085：不输入购买数量［购买］	
Summary： 不输入购买数量，直接购买	
Steps： 注册用户“米奇”查看“诺基亚 N7300”的商品详细信息，未输入购买数量，直接单击［购买］按钮	Expected Results： 提示“您尚未输入购买数量，要购买 1 件该商品吗？”，用户单击［确定］按钮后，向“米奇”的购物车中添加 1 件该商品，否则，不添加到购物车
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 086：在购买数量中输入字母	
Summary： 在购买数量中输入字母	
Steps： 在“购物数量”文本框中输入字母“a”，单击［购买］按钮	Expected Results： 提示“购买数量只允许输入数字，请重新输入”
等价类划分法	
Pass/Fail：	Test Notes：
Author admin	

（2）编写购物车管理的测试用例集

Test Case 087：查看购物车	
Summary： 单击［查看购物车］，可以查看购物车中购买的商品	
Steps： “米奇”，单击［查看购物车］按钮	Expected Results： 进入到“米奇”的购物车查看页面，可以查看已购商品详情
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 088：生成购物单	
Summary： 生成购物单	
Steps： 1．“米奇”确认购物车内的商品无误，单击［生成购物单］ 2．单击［确认］按钮 3．单击［放弃］按钮	Expected Results： 1．系统弹出确认信息，列出购物清单及应支付的总金额 2．提示“购物单已生成，祝您购物愉快” 3．［生成购物单］操作被取消
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 089：清空购物车	
Summary： 清空购物车	
Steps： 1．“米奇”单击［清空购物车］ 2．单击［取消］按钮 3．单击［确定］按钮	Expected Results： 1．提示“您确认要清空购物车吗” 2．“清空购物车”操作被取消 3．购物车被清空
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 090：删除商品	
Summary： 删除商品	
Steps： 1．“米奇”在商品列表中找到“儿童电动车”，单击“儿童电动车”后面的［删除］按钮 2．单击［取消］按钮 3．单击［确定］按钮	Expected Results： 1．提示“您确认要从购物车中删除该商品吗？” 2．删除操作被取消 3．商品被删除
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 091：打印订单	
Summary： 打印订单	
Steps： 1．“米奇”单击［打印订单］按钮 2．单击［打印］按钮	Expected Results： 1．弹出“打印订单界面”，订单中的信息与购物车的内容一致 2．订单从打印机中打出，订单内容与“打印订单界面”中显示的内容一致，页面布局合理
场景法	
Pass/Fail：	Test Notes：
Author admin	

2.3.3 应知应会

1．白盒测试基本概念

白盒测试是用来测试证明每种内部操作和过程是否符合设计规格和要求，又称结构测试或逻辑驱动测试或基于程序的测试。白盒测试技术一般用于单元测试阶段。目前国内很少有公司花很大精力去做白盒测试，商业软件测试技术主要是黑盒测试，白盒测试全由开发人员来完成。

白盒测试主要对程序模块检查如下：

（1）保证一个模块中的所有独立的执行路径至少被使用一次；

（2）对所有逻辑值均需测试 true 和 false；

（3）在循环的上下边界及可操作范围内运行所有循环；

（4）测试内部数据结构以确保其有效性。

正如测试专家 Boris Beizer 所说的："错误潜伏在角落里，聚集在边界上"，而白盒测试更可能发现它。

所以，白盒测试就是知道产品内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都有能按预定要求正确工作，而不顾它的功能，白盒测试的主要方法有逻辑驱动、基路测试等等，主要用于软件验证。

"白盒"法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。"白盒"法是穷举路径测试。在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。贯穿程序的独立路径数是天文数字。但即使每条路径都测试了仍然可能有错误。第一，穷举路径测试决不能查出程序违反了设计规范，即程序本身是个错误的程序。第二，穷举路径测试不可能查出程序中因遗漏路径而出错。第三，穷举路径测试可能发现不了一些与数据相关的错误。

白盒测试的目的就是通过检查软件内部的逻辑结构，对软件中的逻辑路径进行覆盖测试，在程序不同地方设立检查点，检查程序的状态，以确定实际运行状态与预期状态是否一致。测试方法总体上分为静态方法和动态方法。静态分析是不通过执行程序而进行测试的技术；动态分析是当软件系统在模拟的或真实的环境中执行之前、之中和之后，对软件系统行为的分析。

2．白盒测试用例设计方法

白盒测试用例编写方法主要有：

基本路径测试、等价类划分/边界值分析测试、覆盖测试、循环测试、数据流测试、程序插桩测试和变异测试等。

其中基本路径测试和覆盖测试法最常用，也是最基本的设计方法。

白盒测试用例编写的注意事项：

（1）由于测试路径可能非常多，由于时间和资源问题，选出足够多的路径测试；

（2）由于深入到程序编码，通常开发人员协助测试人员书写白盒测试用例。

关于白盒测试用例设计方法在这里就不再详细介绍了，请读者参看其他有关书籍。

3．单元测试

白盒测试技术一般用于单元测试阶段，单元测试的对象是软件设计的最小单位——模块。单元测试的依据是详细设计描述，单元测试应对模块内所有重要的控制路径设计测试用例，

以便发现模块内部的错误。系统内多个模块可以并行地进行测试。

（1）单元测试任务

单元测试任务包括：

① 模块接口测试

模块接口测试是单元测试的基础。只有在数据能正确流入、流出模块的前提下，其他测试才有意义。测试接口正确与否应该考虑下列因素：

a．输入的实际参数与形式参数的个数是否相同；

b．输入的实际参数与形式参数的属性是否匹配；

c．输入的实际参数与形式参数的量纲是否一致；

d．调用其他模块时所给实际参数的个数是否与被调模块的形参个数相同；

e．调用其他模块时所给实际参数的属性是否与被调模块的形参属性匹配；

f．调用其他模块时所给实际参数的量纲是否与被调模块的形参量纲一致；

g．调用预定义函数时所用参数的个数、属性和次序是否正确；

h．是否存在与当前入口点无关的参数引用；

i．是否修改了只读型参数；

j．对全程变量的定义各模块是否一致；

k．是否把某些约束作为参数传递。

如果模块内包括外部输入输出，还应该考虑下列因素：

a．文件属性是否正确；

b．OPEN/CLOSE 语句是否正确；

c．格式说明与输入输出语句是否匹配；

d．缓冲区大小与记录长度是否匹配；

e．文件使用前是否已经打开；

f．是否处理了文件尾；

g．是否处理了输入/输出错误；

h．输出信息中是否有文字性错误；

② 模块局部数据结构测试

检查局部数据结构是为了保证临时存储在模块内的数据在程序执行过程中完整、正确。局部数据结构往往是错误的根源，应仔细设计测试用例，力求发现下面几类错误：

a．不合适或不相容的类型说明；

b．变量无初值；

c．变量初始化或默认值有错；

d．不正确的变量名（拼错或不正确地截断）；

e．出现上溢、下溢和地址异常。

除了局部数据结构外，如果可能，单元测试时还应该查清全局数据（例如 FORTRAN 的公用区）对模块的影响。

③ 模块中所有独立执行通路测试

在模块中应对每一条独立执行路径进行测试，单元测试的基本任务是保证模块中每条语句至少执行一次。此时设计测试用例是为了发现因错误计算、不正确的比较和不适当的控制流造成的错误。此时基本路径测试和循环测试是最常用且最有效的测试技术。计算中常见的

错误包括：

a．误解或用错了算符优先级；

b．混合类型运算；

c．变量初值错；

d．精度不够；

e．表达式符号错。

比较判断与控制流常常紧密相关，测试用例还应致力于发现下列错误：

a．不同数据类型的对象之间进行比较；

b．错误地使用逻辑运算符或优先级；

c．因计算机表示的局限性，期望理论上相等而实际上不相等的两个量相等；

d．比较运算或变量出错；

e．循环终止条件或不可能出现；

f．迭代发散时不能退出；

g．错误地修改了循环变量。

④ 模块的各条错误处理通路测试

一个好的设计应能预见各种出错条件，并预设各种出错处理通路，出错处理通路同样需要认真测试，测试应着重检查下列问题：

a．输出的出错信息难以理解；

b．记录的错误与实际遇到的错误不相符；

c．在程序自定义的出错处理段运行之前，系统已介入；

d．异常处理不当；

e．错误陈述中未能提供足够的定位出错信息。

⑤ 模块边界条件测试

边界条件测试是单元测试中最后，也是最重要的一项任务。众所周知，软件经常在边界上失效，采用边界值分析技术，针对边界值及其左、右设计测试用例，很有可能发现新的错误。

（2）单元测试过程

一般认为单元测试应紧接在编码之后，当源程序编制完成并通过复审和编译检查，便可开始单元测试。测试用例的设计应与复审工作相结合，根据设计信息选取测试数据，将增大发现上述各类错误的可能性。在确定测试用例的同时，应给出期望结果。

应为测试模块开发一个驱动模块（driver）和（或）若干个桩模块（stub），图 2-21 显示了一般单元测试的环境。驱动模块在大多数场合称为“主程序”，它接收测试数据并将这些数据传递到被测试模块，被测试模块被调用后，“主程序”打印“进入-退出”消息。

单元测试环境在搭建的时候注意以下几点：

① 网络连接是否正常；

② 网络流量负担是否过重；

③ 软件测试平台是否可选；

④ 如果 3，是否在不同的软件测试平台进行软件测试；

⑤ 所选软件测试平台的版本（包括 Service Pack）是否正确；

⑥ 所选软件测试平台的参数设置是否正确；

⑦ 所选软件测试平台上正在运行的其他程序是否会影响测试结果；
⑧ 画面的分辨率和色彩设定是否正确；
⑨ 对硬件测试平台的要求和支持程度。

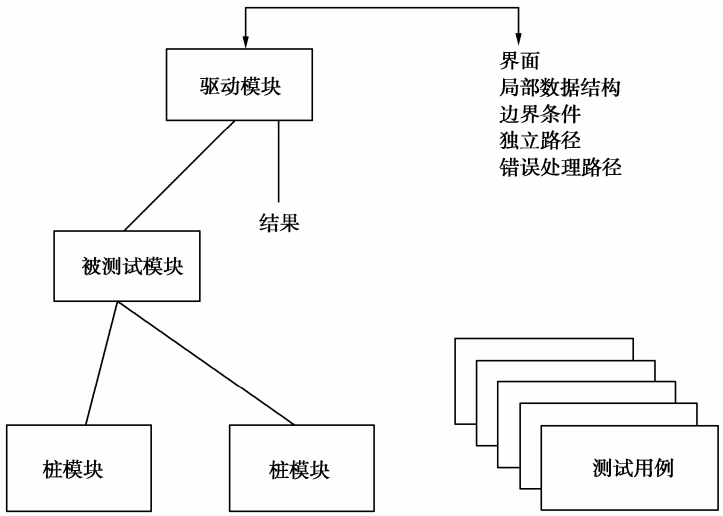


图 2-21　单元测试环境一

为了帮助读者理解，这里再举出另一种单元测试环境的图解，如图 2-22 所示。

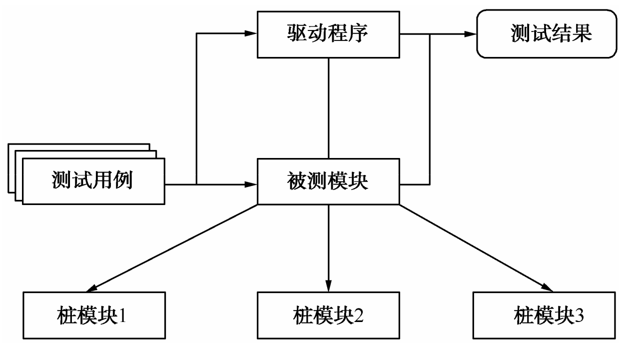


图 2-22　单元测试环境二

驱动模块和桩模块是测试使用的软件，而不是软件产品的组成部分，但它需要一定的开发费用。若驱动和桩模块比较简单，实际开销相对低些。遗憾的是，仅用简单的驱动模块和桩模块不能完成某些模块的测试任务，这些模块的单元测试只能采用下面讨论的综合测试方法。

提高模块的内聚度可简化单元测试，如果每个模块只能完成一个，所需测试用例数目将显著减少，模块中的错误也更容易发现。

4．ALAC（Act-like-a-customer）测试

黑盒测试和白盒测试是最基本的两类软件测试方法，除了前面介绍的这两种测试方法之外，目前又提出一种新的测试方法，即 ALAC(Act-like-a-customer)测试。ALAC 测试是一种基于客户使用产品的知识开发出来的测试方法。ALAC 测试是基于复杂的软件产品有许多错误的原则。最大的受益者是用户，缺陷查找和改正将针对那些客户最容易遇到的错误。图 2-23

就是 ALAC 测试方法的示意图，形象地说明了该方法的含义。

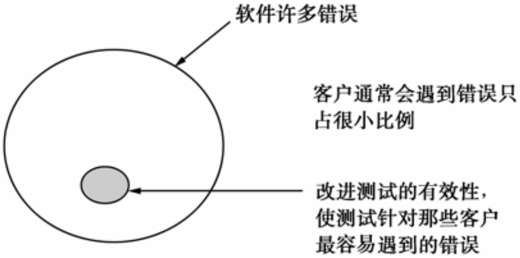


图 2-23　ALAC 测试示意图

2.3.4　拓展任务

尝试进行天天超市购物系统的白盒测试。

工作任务 2.4　Test Suite订单管理

2.4.1　Test Suite订单查询

1．工作任务描述

用户可以即时对某一产品或一些产品下订单，订单信息与客户信息一并提交到数据库等待网站管理员对订单进行处理。管理员可在后台对订单进行实时的处理，例如查询详细的订单如图 2-24 所示。在该页面管理员可以查询某一时间段的订单都订购了哪些商品，为商家时刻了解网店的运营状态提供全面的数据。查询后进入页面如图 2-25 所示，该页面列出订单的详细信息，包括所选购的每种商品的名称、规格、单价和数量。

本节任务是编写查询订单的测试用例。

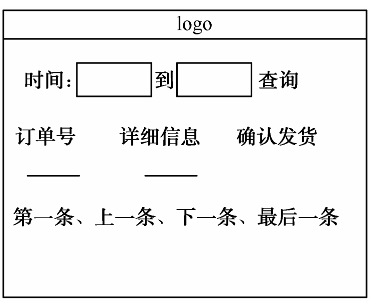


图 2-24　订单查看界面

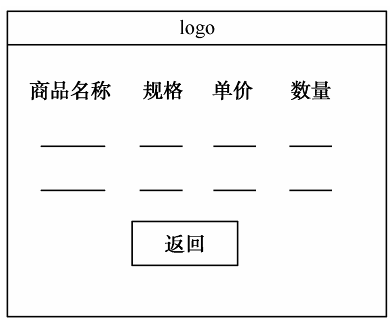


图 2-25　详细订单界面

2．工作过程

编写订单查询的测试用例集：

Test Case 092：按时间段查询
Summary： 按时间段查询

Steps： 管理员在“订单查看界面”的查询区输入 起始时间：2008-12-12 终止时间：2008-12-15 单击［查询］按钮	Expected Results： 仅显示 2008-12-12 日至 2008-12-15 日的所有订单
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 093：起始时间大于结束时间	
Summary： 起始时间大于结束时间	
Steps： 管理员在“订单查看界面”的查询区输入 起始时间：2008-12-18 终止时间：2008-12-15 单击［查询］按钮	Expected Results： 提示“查询条件”“起始时间应小于等于结束时间”
场景法	
Pass/Fail：	Test Notes：
Author admin	

2.4.2 Test Suite订单查看

1．工作任务描述

管理员在如图 2-5 所示的后台管理主页面上单击［查看］按钮进入如图 2-24 所示的订单查看的页面查看客户的订单，在该页面显示了所有客户订单。

本节任务是编写查看订单这一功能的测试用例。

2．工作过程

编写订单查看的测试用例集：

Test Case 094：翻页	
Summary： 单击翻页按钮可以正确跳转到相应的页面上	
Steps： 1．单击［查看］按钮 2．单击［第一条］、［下一条］、［上一条］、［最后一条］	Expected Results： 1．弹出订单查看界面 2．可以正确跳转到对应的页面上，不存在部分信息重复显示，部分信息显示不出来的情况
场景法	
Pass/Fail：	Test Notes：
Author admin	

2.4.3　Test Suite订单详情

1．工作任务描述

管理员进入订单查看页面后，可以看到订单详细情况，确认该订单是否已经处理，如果已经发货，则会弹出已发货的页面如图 2-26 所示，并提供如图 2-27 所示的打印订单的页面。

本节任务就是编写处理订单的测试用例集。

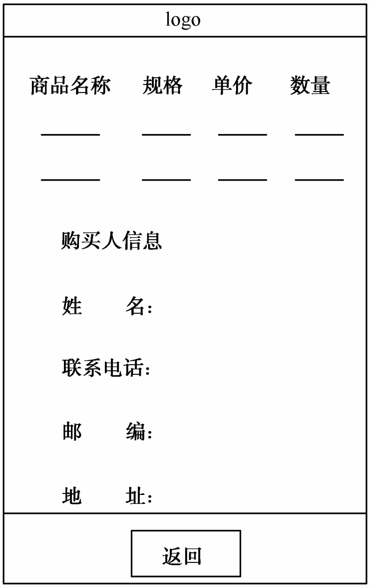


图 2-26　确认发货后界面

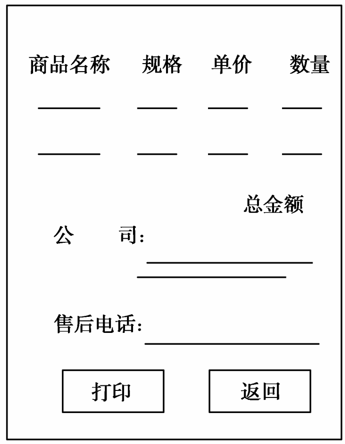


图 2-27　打印订单界面

2．工作过程

编写订单详情的测试用例集：

Test Case 095：查看订单详情	
Summary： 查看订单详情	
Steps： 1．管理员单击［查看］按钮 2．单击订单号是“5505”的［详情］链接文字	Expected Results： 1．进入订单查看界面 2．弹出“5505”的订单详情页面
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 096：确认发货	
Summary： 确认发货	
Steps： 1．管理员单击［查看］按钮 2．单击“订单号”是“5505”的订单后面的链接文字“确认发货”	Expected Results： 1．弹出“订单查看界面” 2．订单“5505”的订单状态变为“已发货”；同时弹出“确认发货后界面”，界面中的内容与“5505”订单的内容一致

场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case097：打印发货详单	
Summary： 打印发货详单	
Steps： 管理员在“确认发货后界面”中单击［打印］按钮	Expected Results： 打印出发货详单，打印页面美观大方，信息内容与“确认发货后界面”中显示的内容一致
场景法	
Pass/Fail：	Test Notes：
Author admin	

2.4.4 应知应会

1．Web系统测试概述

软件大体分为 Web 应用程序（B/S）和 Windows 应用程序（C/S）。随着 Internet 的快速发展，许多传统的信息和数据库系统都被移植到了互联网上，产生了越来越多的 Web 网站，网站的测试也越来越受到重视，网上购物系统就是基于 Web 的应用系统。即使是最简单的一个 Web 应用程序都会包括客户端、中间件和服务器端三个部分，如果要开发一个大型的系统，则包括的内容更复杂。因此，Web 系统测试是一项纷繁复杂的工作，对 Web 系统的测试主要就是对它从全面性、适合性和标准性等多方面进行检查。

基于 Web 的系统测试与传统的软件测试既有相同之处，也有不同的地方，对软件测试提出了新的挑战。基于 Web 的系统测试不但需要检查和验证是否按照设计的要求运行，而且还要评价系统在不同用户的浏览器端的显示是否合适。重要的是，还要从最终用户的角度进行安全性和可用性测试。

2．Web系统测试阶段

测试阶段也叫测试策略。针对 Web 系统的特点，需要对它的各个方面进行测试。按软件的质量特性可以分为功能测试、性能测试、安全性测试、兼容性测试和易用性测试等；按职能来分，可分为应用功能的测试、Web 应用服务的测试、安全系统的测试和数据库服务的测试；按系统架构来分，可分为客户端的测试、服务器端的测试和网络上的测试；按开发阶段来分，可分为 Web 应用设计测试、Web 应用开发测试和 Web 应用运行测试。由于本书着重讲解黑盒测试，所以在前面详细介绍了功能测试的基础上，工作任务 2.5 中将详细介绍 Web 应用运行测试中的其他测试部分，包括性能测试、链接测试、导航测试、界面测试、兼容性测试、帮助文档测试这六个方面。

总的来说，系统测试共有十六个测试策略，它们是功能测试、性能测试、压力测试、容量测试、安全性测试、可用性测试、GUI 测试、安装测试、配置测试、异常测试、备份测试、健壮性测试、文档测试、在线帮助测试、网络测试、稳定性测试。在实际项目中，可以视情况选择一部分测试策略进行测试。

3．基于Web的系统测试综述

基于 Web 的系统开发，如果缺乏严格的过程控制，在开发、发布、实施和维护过程中可能会发生一些严重的问题，失败的可能也非常大。如果 Web 系统很复杂，项目的失败肯定会引发诸多的问题，从而引起 Web 软件危机。一般的软件的发布周期以月或以年计算，而 Web 系统的发布周期以天计算甚至以小时计算。Web 测试人员必须以更短的时间发布系统，测试人员和测试管理人员面临着从测试传统的 C/S 结构和框架环境到测试快速改变的 Web 应用系统的转变。

下面从 Web 系统测试的功能测试、性能测试、可用性测试、客户端兼容性测试、安全性测试等方面讨论。

（1）功能测试

① 链接测试

链接是在页面之间切换和指导用户到一些不知道网址的页面的主要方式，它是 Web 应用系统的主要和明显特征。可以从以下几个方面考虑：

a．测试所有链接是否按指示的那样确实链接到了该链接的页面；

b．测试所链接的页面是否存在；

c．保证 Web 系统上没有孤立的页面。

其中，所谓孤立页面是指没有链接指向该页面，只有知道正确的 URL 地址才能访问。

链接测试可以采用测试工具进行自动化测试。链接测试发生的时间是在集成测试阶段完成，即在整个 Web 系统的所有页面开发完成之后进行。

② 表单测试

表单操作是在用户向 Web 系统管理员提交信息时进行的。最常见的比如用户注册、登录等。表单测试是用来测试用户提交操作的完整性，以校验提交给服务器的信息的正确性。可以从以下几个方面考虑：

a．比如用户提交的出生日期是否符合常理，填写的所属省份与所在城市是否匹配等。

b．用户所填写的信息是否在表单可以接受的值的范围。如果不接受，系统是否会报出错误提示。

③ Cookies 测试

Cookies 是用来存储用户信息和用户在应用系统中的操作。当用户使用 Cookies 访问一个应用系统时，Web 服务器会发送关于用户的信息，把该信息以 Cookies 的形式存储在客户端计算机上，这可用来创建动态和自定义页面或者存储登录等信息。

Cookies 测试可从以下几个方面考虑：

a．Cookies 是否起作用；

b．Cookies 是否按预定的时间进行保存；

c．刷新对 Cookies 有什么影响。

④ 设计语言测试

设计语言指开发该 Web 系统所使用的语言，设计语言测试就是用来测试设计 Web 系统的不同语言版本的兼容性。因为一个系统不可能由一个人来完成，所以开发人员设计系统时所采用的语言版本之间可能会存在不兼容的问题。

设计语言测试可以从以下几个方面考虑：

a．所使用的 HTML 的版本；

b．脚本语言的版本。比如：Java、JavaScript、ActiveX、VBScript 或 Perl 等脚本语言。

⑤ 数据库测试

数据库测试顾名思义就是对系统所使用的数据库进行测试。在当前的 Web 系统中，一般情况下都会使用数据库为系统的管理、运行、查询和实现用户对数据存储的请求等提供空间。

数据库测试可以从以下两方面进行考虑：

a．数据一致性。这种错误主要是由于用户提交的表单信息不正确造成的。

b．输出错误。这种错误主要是由于网络速度和程序设计的问题引起的。

（2）性能测试

① 连接速度测试

连接速度指用户打开一个页面的快慢程度，很多时候是受用户上网方式的影响的。

连接速度测试可以从以下几个方面进行考虑：

a．访问页面时用户需要等待的时间长短。一般不要超过 5 秒钟，如果系统响应时间太长，用户可能就会因为没有耐心等待而离开页面。

b．页面有超时限制时，响应速度会不会慢到用户没来得及浏览内容而因为超时需要重新登录系统。

c．连接速度会不会太慢而导致页面数据丢失，用户得不到真实的页面。

② 负载测试

负载指的是某一时刻同时访问 Web 系统的用户数量，或者是在线数据处理的数量。负载测试是为了测量 Web 系统在某一负载级别上的性能，以保证 Web 系统在需求范围内能正常工作。

负载测试可以从以下几个方面进行考虑：

a．Web 系统可以允许多少个用户同时在线；

b．超出上面所描述的数量后，系统会发生什么现象；

c．Web 系统能否处理大量用户对同一个页面的同时请求。

负载测试应该在 Web 系统发布后，在实际的网络环境中进行。

③ 压力测试

压力测试是指系统的限制和故障恢复能力。是用来测试 Web 系统会不会崩溃，以及在什么情况下会崩溃。如果系统能承受的压力不够大时，黑客会利用系统的这一缺陷提供错误的数据负载，使 Web 系统崩溃。

压力测试的范围包括表单、登录和信息传输的页面等。

（3）可用性测试

① 导航测试

导航描述用户在一个页面内的操作方式。Web 应用系统的用户趋向于目的驱动，快速地扫描一个 Web 系统有没有自己需要的或感兴趣的信息，如果没有捕捉到就会很快离开转向别的系统。尤其是在生活快节奏的今天，很少有用户愿意花时间去熟悉 Web 应用系统的结构，所以 Web 应用系统导航对用户的帮助要尽可能地准确。

Web 应用系统最好在层次决定后就进行测试，如果让最终用户参加到导航测试中，效果会更加明显。

导航测试可以从以下几个方面进行考虑：

a．导航是否直观；

b．Web 系统的主要部分是否可通过主页存取；

c．Web 系统是否需要站点地图、搜索引擎或其他的导航帮助。

② 图形测试

图形指 Web 应用系统页面上包括的图片、动画、边框、颜色、字体、背景及按钮等。一个 Web 系统如果使用适当的这些多媒体信息，可以使页面增辉不少。

图形测试主要从以下几个方面进行考虑：

a．要确保页面上所使用的图形有明确的用途，而不是将图片或动画胡乱地堆砌在一起，这样既浪费传输速度又影响页面的美观。

b．验证所有的页面风格是否一致。

c．背景颜色和字体颜色及前景颜色是否搭配。

d．图片的大小和质量也需要认真考虑，一般采用 JPG 或 GIF 格式的图片。

③ 内容测试

内容测试是用来检验 Web 应用系统所提供信息的正确性、准确性和相关性。内容测试也可以从这三方面入手。

a．正确性指信息是可靠的还是误传的。

b．准确性指信息是否有语法或拼写错误。

c．相关性指是否在当前页面可以找到与当前浏览信息相关的信息列表或入口。

④ 整体界面测试

整体界面指整个 Web 应用系统整体的的页面结构设计。整体界面测试是调查用户对界面的整体风格的评价。一般 Web 应用系统采取在主页上做问卷调查的形式进行的。这种测试也最好是由最终用户进行真实参与进行。

整体界面测试可以从以下几方面进行考虑：

a．用户浏览 Web 系统时是否感觉舒适。

b．用户能否凭直觉就知道要找的信息在什么位置。

c．整个 Web 系统的设计风格是否一致。

（4）客户端兼容性测试

① 平台测试

目前市场上有很多种不同类型的操作系统，比如 Windows、UNIX、Linux、Macintosh 等。Web 应用系统的最终用户使用哪种类型的操作系统，取决于用户系统的配置。这样就会引发兼容性问题，同一个系统可能在某些操作系统下能正常运行，而在另外一些操作系统中运行失败。因此就有必要在 Web 系统发布之前对它进行平台测试。

平台测试主要就是测试 Web 系统能在什么类型的操作系统上正常运行，在哪些操作系统中会运行失败。

② 浏览器测试

Web 应用系统最核心的构件就是浏览器，浏览器的版本很多，它们对设计语言的规格支持度不同，所以要进行浏览器的兼容性测试。

浏览器测试可以从以下几个方面进行考虑：

a．不同厂商的浏览器对 Java、JavaScript、ActiveX、plug-ins 或 HTML 的支持不同。

b．框架和层次结构风格在不同厂商的浏览器中的显示不同。

c．不同厂商的浏览器对安全性和 Java 的设置不同。

（5）安全性测试

安全性测试是为了保证用户访问Web应用系统的安全性。可以从以下几个方面进行考虑：

a．现在的 Web 应用系统基本采用先注册，后登录的方式。因此，必须测试有效和无效的用户名和密码，要注意到是否大小写敏感，可以试多少次的限制，是否可以不登录而直接浏览某个页面等。

b．Web 应用系统是否有超时的限制，也就是说，用户登录后在一定时间内（例如 15 分钟）没有单击任何页面，是否需要重新登录才能正常使用。

c．为了保证 Web 应用系统的安全性，日志文件是至关重要的。需要测试相关信息是否写进了日志文件、是否可追踪。

d．当使用了安全套接字时，还要测试加密是否正确，检查信息的完整性。

e．服务器端的脚本常常构成安全漏洞，这些漏洞又常常被黑客利用。所以，还要测试没有经过授权，就不能在服务器端放置和编辑脚本的问题。

Web 应用系统与 Windows 应用系统有相同和不同的地方，读者可以结合 Web 系统测试的介绍比较二者的异同之处，进一步学习测试知识。

2.4.5　拓展任务

结合天天超市管理系统，学习 Windows 应用程序测试。

工作任务 2.5　Test Suite其他测试

2.5.1　Test Suite性能测试

1．工作任务描述

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行的测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

因为性能测试不同于平时的测试用例，尽可能把性能测试用例设计的复杂，才有可能发现软件的性能瓶颈。

2．工作过程

编写性能测试的测试用例集：

<table>
<tr><td colspan="2">Test Case 098：大数据量测试</td></tr>
<tr><td colspan="2">Summary：
注册、登录、查看商品、购买商品、商品类别维护、商品维护、订单维护
业务按照 3∶6∶20∶4∶1∶2∶4 的比例进行混合加压
具备 200 个业务管理员，2 万个注册用户，800000 万条历史数据，考察业务是否能正常运行</td></tr>
<tr><td>Steps：
并发用户数 2000</td><td>Expected Results：
考察系统是否可以正常运行</td></tr>
</table>

续表

大数据量法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 099：负载测试	
Summary： 注册、登录、查看商品、购买商品、商品类别维护、商品维护、订单维护 业务按照 3∶6∶20∶4∶1∶2∶4 的比例进行混合加压	
Steps： 以每分钟增加 50 个并发用户	Expected Results： 考查在用户响应时间<5 秒的情况下，系统支持的最大并发用户数
负载测试法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 100：疲劳强度测试	
Summary： 注册、登录、查看商品、购买商品、商品类别维护、商品维护、订单维护 业务按照 3∶6∶20∶4∶1∶2∶4 的比例进行混合加压	
Steps： 并发用户数 2000	Expected Results： 考察系统可以无故障运行多长时间
疲劳强度测试法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 101：压力测试	
Summary： 注册、登录、查看商品、购买商品、商品类别维护、商品维护、订单维护 业务按照 3∶6∶20∶4∶1∶2∶4 的比例进行混合加压	
Steps： 以每分钟增加 50 个并发用户	Expected Results： 考查在服务器 CPU 使用率达到 85%，内存使用率达到 90%时，系统可以支持的最大并发用户数
压力测试法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 102：按钮状态是否正确	
Summary： 与正在进行的操作无关的按钮应该加以屏蔽	

Steps： 检测运行程序的过程中，与正在进行的操作无关的按钮是否加以屏蔽	Expected Results： 与正在进行的操作无关的按钮不显示或者置为灰色（处于不可用状态）
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 103：按钮的摆放位置是否合理	
Summary： 错误使用容易引起界面退出或关闭的按钮不应该放在容易单击的位置	
Steps： 检测各个页面中的按钮摆放位置	Expected Results： 错误使用容易引起界面退出或关闭的按钮不应该放在横排开头、横排最后及竖排最后
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 104：重要按钮的摆放位置是否合适	
Summary： 重要的命令按钮与使用较频繁的按钮要放在界面上醒目的位置	
Steps： 检查各个页面重要按钮的摆放位置是否合适	Expected Results： 重要的命令按钮与使用较频繁的按钮放在了界面上醒目的位置上
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 105：关闭错误提示后的光标定位	
Summary： 关闭用户输入错误的提示信息后，光标应定位到对应的输入框中	
Steps： 关闭提示信息 “××输入域不能为空” “××信息已存在” “××字段输入的数据类型不正确”	Expected Results： 光标应定位到“××”输入框中
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 106：非法访问	
Summary： 未登录直接访问	
Steps： 复制需要登录后才可以访问的页面的 URL， 在未登录的情况下，将 URL 复制到地址栏，单击［转到］按钮	Expected Results： 提示“您尚未登录！”
场景法	
Pass/Fail：	Test Notes：
Author admin	

2.5.2　Test Suite链接测试

1．工作任务描述

链接是 Web 网站的一个主要特征，它是在页面之间切换和引导用户去一些未知地址页面的主要手段。

链接测试的内容：

（1）测试所有链接是否按指示的那样确实链接到了应该链接的页面；

（2）测试所链接的页面是否存在；

（3）保证 Web 网站上没有孤立的页面。所谓孤立页面是指没有链接指向该页面，只有知道正确的 URL 地址才能访问。

（4）链接测试可以手动进行，也可以自动进行。

（5）链接测试必须在集成测试阶段完成，也就是说，在整个 Web 网站的所有页面开发完成之后进行链接测试。

2．工作过程

编写链接测试的测试用例集：

Test Case 107：所有链接均链接到了该链接的页面	
Summary： 测试所有链接是否按指示的那样确实链接到了该链接的页面	
Steps： 单击页面中的每一个链接，检查链接是否按照指示的那样确实链接到了该链接的页面	Expected Results： 所有链接均链接到了该链接的页面
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 108：链接的页面不存在	
Summary： 测试所链接的页面是否存在	
Steps： 单击每一个链接，检查所链接的页面是否存在	Expected Results： 所有链接均有链接页面

错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 109：系统上没有孤立的页面	
Summary： 保证 Web 应用系统上没有孤立的页面，所谓孤立页面是指没有被链接的页面	
Steps： 使用测试工具 XENU，检测系统	Expected Results： 系统上没有孤立的页面
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

2.5.3 Test Suite导航测试

1．工作任务描述

导航是在不同的用户接口控制之间，例如按钮、对话框、列表和窗口等；或在不同的连接页面之间，导航描述了用户在一个页面内操作的方式。

导航测试的内容有：

（1）导航是否直观？

（2）Web 系统的主要部分是否可以通过主页访问？

（3）Web 系统是否需要站点地图、搜索引擎或其他的导航器帮助？

（4）测试 Web 系统的页面结构。

（5）导航条、菜单、连接的风格是否一致？

（6）各种提示是否准确，确保用户凭直觉就知道是否还有内容，内容在什么地方。

最好让最终用户参与导航测试，效果将更加明显。

2．工作过程

编写导航的测试用例集：

Test Case 110：导航直观	
Summary： 导航按钮清晰可见，便于使用	
Steps： 检查各个页面中的导航按钮	Expected Results： 导航按钮清晰可见，便于使用
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 111：主页中是否提供了主要模块的链接	
Summary： 系统中的主要模块应该可以通过主页链接，直接访问	
Steps： 检测主页中是否包含了所有主要模块的链接	Expected Results： 在前、后台主页面可以直接进入到目标模块（如：商品类别管理、商品管理、订单管理、个人信息维护、购物车管理模块）
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case112：是否有导航帮助功能	
Summary： 有站点地图、搜索引擎或其他的导航帮助	
Steps： 检测页面中是否提供站点地图、搜索引擎及其他的导航帮助	Expected Results： 在页面左侧或上方提供站点地图、搜索引擎及导航帮助信息
Pass/Fail：	Test Notes：
Author admin	

Test Case 113：导航能否流动到目的地	
Summary： 按照导航信息，应能顺利完成各项任务	
Steps： 按照导航信息能否顺利完成购物、查看购物车、生成订货单等的业务操作	Expected Results： 按照导航可以一步一步地顺利完成购物过程中的各种操作
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

2.5.4　Test Suite界面测试

1．工作任务描述

整体界面测试是对整个Web系统的页面结构设计的测试，是用户对系统的一个整体感受。例如，当用户浏览Web网站时，应考虑是否感到舒适？是否凭直觉就知道要找的信息在什么地方？整个Web应用系统的设计风格是否一致？

2．工作过程

编写界面测试的测试用例集：

Test Case 114：调整浏览器大小，页面还能完全显示
Summary： 调整浏览器大小，页面还能完全显示

Steps： 拖动浏览器边框，调整浏览器大小	Expected Results： 当拖动到一定大小后，不能再缩小或放大。页面始终显示完全
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 115：提示、警告、或错误说明应该清楚、明了、恰当	
Summary： 提示、警告、或错误说明应该清楚、明了、恰当	
Steps： 测试过程中，关注系统弹出的提示、警告、或错误说明	Expected Results： 提示、警告、或错误说明应该清楚、明了、恰当
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 116：是否有错误提示	
Summary： 对运行过程中出现问题而引起错误的地方要有提示，避免形成无限期的等待	
Steps： 检测对运行过程中出现问题而引起错误的地方是否有提示	Expected Results： 出现问题而引起错误的地方有提示信息，让用户明白错误出处，避免用户无限期的等待
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 117：是否有提示说明	
Summary： 非法的输入或操作应有足够的提示说明	
Steps： 检测系统对非法输入和执行非法操作是否给出了提示说明	Expected Results： 给出清晰、明确的说明信息
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 118：是否提供放弃的选择项	
Summary： 对可能造成数据无法恢复的操作必须提供确认信息,给用户放弃选择的机会	
Steps： 检测系统运行过程中，对于删除、清空、修改等无法恢复的操作，是否提供确认信息,给用户放弃选择的机会	Expected Results： 提供了确认信息,用户可以选择放弃

错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 119：所有页面字体的风格一致	
Summary： 验证所有页面字体的风格是否一致	
Steps： 检查所有页面字体的风格	Expected Results： 所有页面字体的风格一致
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 120：背景颜色与字体颜色和前景颜色相搭配	
Summary： 检查背景颜色与字体颜色和前景颜色是否相搭配	
Steps： 检查所有页面的背景颜色与字体颜色和前景颜色是否相搭配	Expected Results： 背景颜色与字体颜色和前景颜色相搭配
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 121：表格里的文字折行显示	
Summary： 表格里的文字是否都有折行	
Steps： 查看表单中一行无法显示完全的数据，是否折行显示	Expected Results： 表格里的文字折行显示
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 122：窗体布局	
Summary： 窗体布局是否合理、结构是否清晰	
Steps： 检查所有页面的窗体布局	Expected Results： 窗体布局合理、结构清晰
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 123：页面中的说明文字	
Summary： 页面中的说明文字，语句通顺、语义明确	
Steps： 检查各个页面的说明文字	Expected Results： 语句通顺、语义明确
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 124：检查拼写错误	
Summary： 检查页面中是否有拼写错误	
Steps： 检查各个页面中是否有拼写错误	Expected Results： 页面中无拼写错误
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

2.5.5 Test Suite兼容性测试

1．工作任务描述

软件兼容性测试是指检查软件之间是否正确地交互和共享信息。交互可以在同时运行于同一台计算机上，甚至在相隔几千公里通过英特网连接的不同计算机上的两个程序之间进行。交互还可以简化为在软件上保存数据，然后拿到其他房间的计算机上。

如果受命对新软件进行兼容性测试，就需要解答以下问题：

（1）软件设计要求与何种其他平台（操作系统、Web 浏览器或者操作环境）和应用软件保持兼容？如果要测试的软件是一个平台，那么设计要求什么应用程序在其上运行？

（2）应该遵守何种定义软件之间交互的标准或者规范？

（3）软件使用何种数据与其他平台和软件交互和共享信息？

兼容性测试实质上是一项庞大而又复杂的任务，如果测试对象是新操作系统，就可能只要求对字处理程序和图形程序进行兼容性测试。如果测试对象是应用程序，就可能要求在多个不同的平台上进行兼容性测试。

2．工作过程

编写兼容性测试的测试用例集：

Test Case 125：分辨率测试	
Summary： 测试系统在不同分辨率下是否能够正常显示	
Steps： 1．在浏览器的地址栏中输入访问“网上购物系统”的 url，单击［转到］按钮 2．右键单击操作系统的桌面；选择［属性］\［设置］，调整［屏幕分辨率］，单击［确定］按钮，保存所做的修改 3．切换到购物系统的各个页面 4．重复执行第 2 和第 3 步骤	Expected Results： 1．弹出“网上购物系统”主页 2．分辨率改变 3．所有页面均能正常显示，页面美观、控件间的相对位置合理 4．页面在所有分辨率下均能正常显示，页面美观、控件间的相对位置合理

错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 126：浏览器测试	
Summary： 系统在所有主流的浏览器（IE 6、IE 7、遨游、火狐、腾迅、360 等）下均能正常使用	
Steps： 1．用户使用不同的主流的浏览器（如：IE 6、IE 7、遨游、火狐、腾迅、360 等），在地址栏中输入“网上购物系统”URL 2．在购物网站的不同页面间切换	Expected Results： 1．可以顺利地进入到“网上购物系统”主页面（至少支持 IE 6 和 IE 7） 2．所有的功能均可用，且页面美观
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 127：平台测试	
Summary： 系统可以搭建在 Windows、UNIX、Macintosh、Linux 等操作系统上，数据库可以移植到 Oracle、SQL Server、Sybase 上，应用服务器可以使用 Tomcat、Websphere 上	
Steps： 将系统搭建在不同操作系统、数据库、应用服务器上	Expected Results： 系统至少支持 2 种以上运行环境
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 128：打印机	
Summary： 使用各种类型的打印机均能打印出相关单据，单据完整、清晰、页面美观	
Steps： 安装各种类型的打印机，单击［打印］按钮	Expected Results： 可以打印出相关表单，表单内容完整、页面布局合理、美观大方（至少支持主流打印机）
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

2.5.6 Test Suite帮助文档测试

1．工作任务描述

文档测试是一种很重要的阶段，一般可以叫评审。就是通过对需求分析文档，概要设计文档，详细设计文档等的阅读，找出错误的或者不合适的地方。

因为软件测试技术发展到现在，软件测试不仅仅是开发完再测试，而是要深入到软件开

发流程中，从最初的需求分析入手，一直贯穿到软件开发完毕，甚至到维护。

2．工作过程

编写帮助文档测试的测试用例集：

<table>
<tr><td colspan="2">Test Case 129：帮助文档中的性能介绍与说明要与系统性能配套一致</td></tr>
<tr><td colspan="2">Summary：
帮助文档中的性能介绍与说明要与系统性能配套一致</td></tr>
<tr><td>Steps：
单击［帮助］，打开帮助文档</td><td>Expected Results：
帮助文档中的性能介绍与说明要与系统性能配套一致</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 130：打包新系统时，对做了修改的地方在帮助文档中要做相应的修改</td></tr>
<tr><td colspan="2">Summary：
打包新系统时，对做了修改的地方在帮助文档中要做相应的修改</td></tr>
<tr><td>Steps：
单击［帮助］，打开最新的帮助文档</td><td>Expected Results：
做了修改的地方在帮助文档中已做相应的修改</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 131：提供搜索功能</td></tr>
<tr><td colspan="2">Summary：
用户可以用关键词在帮助索引中搜索所要的帮助，也可以使用帮助主题词定位到主题</td></tr>
<tr><td>Steps：
单击［帮助］</td><td>Expected Results：
在帮助页面中用户可以用关键词在帮助索引中搜索所要的帮助，也可以通过帮助主题词定位到所关心的主题</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

<table>
<tr><td colspan="2">Test Case 132：提供技术支持方式</td></tr>
<tr><td colspan="2">Summary：
在帮助中应该提供我们的技术支持方式，一旦用户难以自己解决可以方便的寻求新的帮助方式</td></tr>
<tr><td>Steps：
单击［帮助］</td><td>Expected Results：
在帮助页面可以很容易地找到技术支持方式
一旦用户难以自己解决可以方便的寻求新的帮助方式</td></tr>
<tr><td colspan="2">错误推测法</td></tr>
<tr><td>Pass/Fail：</td><td>Test Notes：</td></tr>
<tr><td colspan="2">Author admin</td></tr>
</table>

Test Case 133：提供留言功能	
Summary： 在帮助中应提供留言管理功能，让用户可以方便进行沟通	
Steps： 单击［帮助］	Expected Results： 提供留言功能，方便在线用户进行经验交流
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 134：返回	
Summary： 单击［返回］按钮，可以返回到上一页面	
Steps： 在网上购物系统的不同页面间切换，单击［返回］按钮	Expected Results： 可以返回到上一页面
场景法	
Pass/Fail：	Test Notes：
Author admin	

Test Case 135：已查看过的商品是否被突出显示	
Summary： 查看和未查看过的商品，在显示上应有所区别	
Steps： “米奇”单击商品“儿童电动车”，查看商品详情，单击［返回］按钮，返回到商品浏览页面	Expected Results： 已被查看过的“儿童电动车”与其他未被查看过的商品在显示上有所区别（例如：已被访问过的字体是褐色）
错误推测法	
Pass/Fail：	Test Notes：
Author admin	

2.5.7 应知应会

如前所述，系统测试种类繁多，最多可达三十多种。如此纷繁复杂的测试种类，初学者肯定会感觉无从着手，表 2-10 是对测试类型的分析与总结。

表 2-10 系统测试设计——测试类型分析

测 试 类 型	测试关注点	完 成 情 况
	用户层	
用户支持测试	用户手册、使用帮助、支持客户的其他产品技术手册是否正确、是否易于理解、是否人性化	
用户界面测试	测试对象控件或访问入口正确，符合用户需求	
	界面风格统一，界面美观、直观	
	操作友好、人性化	
	易操作性	

<table>
<tr><th>测 试 类 型</th><th>测试关注点</th><th>完 成 情 况</th></tr>
<tr><td colspan="3">用户层</td></tr>
<tr><td>可维护性测试</td><td>系统软、硬件实施和维护功能的方便性</td><td></td></tr>
<tr><td rowspan="5">安全性测试</td><td>操作安全性
（注：核实只有具备系统和应用程序访问权限的主角才能访问系统和应用程序；核实主角只能访问其所属用户类型已被授权使用的那些功能操作）</td><td></td></tr>
<tr><td>数据安全性
（注：关注数据访问的安全性，防止交易敏感数据不被第三方截获、窃取、篡改和伪造。测试内容为：数据加密，安全通信，安全存储）</td><td></td></tr>
<tr><td>网络安全测试
（注：该层次的测试主要是为了防止黑客的恶意攻击和破坏，如：病毒，DOS 攻击等。测试的方式主要是模拟黑客对系统进行入侵攻击，然后对攻击的结果进行分析，并逐步完善系统的安全性能）</td><td></td></tr>
<tr><td>安全认证测试
（注：确保交易双方不被其他人冒名顶替。测试内容为安全认证）</td><td></td></tr>
<tr><td>安全交易协议测试
（注：有效避免交易双方出现互相抵赖的情况）</td><td></td></tr>
<tr><td colspan="3">应用层</td></tr>
<tr><td rowspan="6">性能测试</td><td>并发性能测试
（注：并发用户操作下，不断增加并发用户数量，分析系统性能指标、资源状况。主要关注点：交易结果、每分钟交易数、交易响应时间（最小服务器响应时间、平均服务器响应时间、最大服务器响应时间））</td><td></td></tr>
<tr><td>压力测试
（注：不断对系统施压，通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试）</td><td></td></tr>
<tr><td>强度测试
（注：系统在极限或异常资源情况下，即系统资源处于特别低的状况下，软件系统运行情况确定系统综合交易指标和资源监控指标，保证系统能否在规格强度下运行）</td><td></td></tr>
<tr><td>负载测试
（注：关注各种工作负载情况下的性能指标，测试当负载逐渐增加到超负荷状态时，系统组成部分的相应输出项，例如通过量、响应时间、CPU 负载、内存使用等来决定系统的性能）</td><td></td></tr>
<tr><td>疲劳测试
（注：采用系统稳定运行情况下能够支持的最大并发用户数，持续执行一段时间业务，通过综合分析交易执行指标和资源监控指标来确定系统处理最大工作量强度性能的过程）</td><td></td></tr>
<tr><td>大数据量测试
（注：针对某些系统存储、传输、统计、查询等业务分别进行大数据量测试）</td><td></td></tr>
</table>

测 试 类 型	测试关注点	完 成 情 况
应用层		
性能测试	容量测试 （注：确定系统可处理同时在线的最大用户数）	
	破坏性测试 （注：超出系统能承受的压力点后，系统出现错误状态、出现错误比率及恢复能力；对软件进行异常的操作，如：删除配置文件）	
系统可靠性、稳定性测试	可靠性测试 （注：主要测试系统在负载压力下系统运行是否正常）	
	稳定性测试 （注：确保系统在长期使用周期内能够在要求的性能指标下正常工作）	
系统兼容性测试	操作系统兼容性 （注：win9x/win2k/winXP/UNIX/Linux……）	
	浏览器兼容性 （注：IE4/IE5/IE6/Firefox/MyIE/TT……）	
	其他支持软件/平台/文件/数据/接口的兼容性	
系统组网测试		
系统安装升级测试	初次安装	
	更新（注：以前安装过相同版本）	
	升级（注：以前安装过较早版本）	
功能层		
功能性测试	初验测试 （注：系统核心功能、基本业务流程的验证）	
	业务场景测试 （注：模拟用户实际操作的业务场景，遍历主要业务流程和业务规则）	
	业务功能的覆盖 （注：关注需求规格定义的所有功能系统是否都已实现）	
	业务功能的分解 （注：将每个功能分解成测试项。关注每个测试项的测试类型都被测试通过）	
	业务功能的组合 （注：相关联的功能项的组合功能都被正确实现）	
	业务功能的冲突 （注：业务功能间存在的功能冲突情况，均测试通过。例如：共享资源访问等）	
	异常处理及容错性 （输入异常数据、或执行异常操作后，系统容错性及错误处理机制的健壮性。例如：重复单击“提交”按钮，提交申请单）	

续表

测 试 类 型	测试关注点	完 成 情 况
子系统层		
	单个子系统的性能 （注：应用层关注的是整个系统各种软、硬件、接口配合情况下的整体性能，这里关注单个系统。）	
	子系统间的接口瓶颈 （例如：子系统间通信请求包的并发瓶颈）	
	子系统间的相互影响 （注：子系统的工作状态变化对其他子系统的影响）	
协议/指标层		
	协议一致性测试	
	协议互通测试	
其他测试类型		
BVT	构建验证测试	
Ad hoc Test	随机测试	
Exploratory Test	探索性测试	
回归测试		

1．性能测试

性能测试即测试软件处理事务的速度，一是为了检验性能是否符合需求，二是为了得到某些性能数据供人们参考（例如用于宣传）。性能测试类型包括负载测试、压力测试和容量测试等。负载测试是用来测试数据在超负荷环境中运行，程序是否能够承担。强度测试是用来测试在系统资源特别低的情况下，软件系统的运行情况。容量测试是用来确定系统可处理同时在线的最大用户数。

性能测试是一项综合性很强的工作，甚至可以作为一项工程来看待。中国软件评测中心将性能测试概括为三个方面：应用在客户端性能的测试、应用在网络上性能的测试和应用在服务器端性能的测试。通常情况下，三方面有效、合理的结合，可以达到对系统性能全面的分析和瓶颈的预测。

林锐博士提出，性能测试的一些注意事项：

（1）不要试图让人拿着钟表去测时间，应当编写一段程序用于计算时间以及相关数据。

（2）应当测试软件在标准配置和最低配置下的性能。

（3）为了排除干扰，应当关闭那些消耗内存、占用 CPU 的其他应用软件（如杀毒软件）。

（4）不同的输入情况会得到不同的性能数据，应当分档记录。例如传输文件的容量从 100kB 到 1MB 可以分成若干等级。

（5）由于环境的波动，同一种输入情况在不同的时间可能得到不同的性能数据，可以取其平均值。

根据不同的工程和项目，我们所选用的度量，及评估方法也有不同之处。但一般情况下，完成性能测试项目的可以参考如下的通用步骤：

（1）制定目标和分析系统
（2）选择测试度量的方法
（3）学习的相关技术和工具
（4）制定评估标准
（5）设计测试用例
（6）运行测试用例
（7）分析测试结果

要做好一件事情，选择合适的工具去完成是很重要的。要进行性能测试，有很多种工具可供选择，比如 LoadRunner、OpenSTA、Microsoft Web Application Stress Tool 等。Matt Maccaux 在 2005 年发表的著作 Approahes to Performance Testing 中，描述了使用工具进行服务器负载测试的结果描述，如图 2-28 所示：

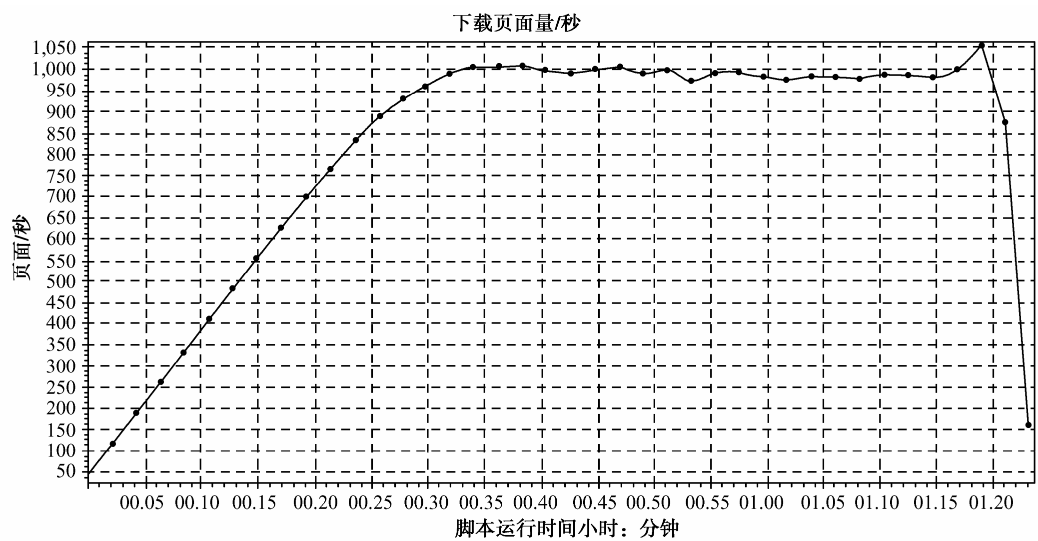


图 2-28　随着负载的增加，系统吞吐量的曲线（单位：页面/秒）

选用的测试工具对测试结果可能会产生很大的影响。假设测试的两个指标是服务器的响应时间和吞吐量，二者都会受到服务器上的负载的影响。而服务器上的负载又受两个因素的影响，同时与服务器通信的用户的数目以及每个用户请求之间的考虑时间的长短。显然，与服务器通信的用户越多，负载就越大。同样，请求之间的考虑时间越短，负载也越大。这两个因素的不同组合会产生不同等级的服务器负载。如图 2-28 所示，随着服务器上负载的增加，吞吐量也会随之上升，并以稳定的速度增长，直到达到一定高度后，在某一个点上稳定下来。如果这个时候服务器上的负载再增加，则吞吐量会急速地下降。

2．链接测试

什么是链接呢？链接是 Web 应用系统的一个主要特征，它是在页面之间切换和指导用户去一些不知道地址的页面的主要手段。

链接测试就是从待测的项目根目录开始，对所有页面中的超链接、图片、页面内部链接等所有的链接进行测试，测试链接是否正确链接到指定的页面，这种情况下，测试工具或程序是无法判断的，所以需要人工进行推测所链接的页面是否正确。如果网站内文件不存在、指定文件链接不存在、指定页面不存在或者存在孤立页面，则要将该链接具体位置及测试结

果记录下来，一直到所有链接测试都结束，并写出测试报告。

链接测试的工具也有很多，比如 Xenu Link Sleuth、HTML Link Validator、Web Link Validat 等等。每种工具都有优劣，在实际项目测试过程中可以取其优避其劣灵活运用。

总之链接测试是在集成测试阶段进行的，即整个应用系统的所有页面完成之后进行。链接测试技术含量不是太高，但却非常重要。一个软件如果出现链接错误，其可用性会大打折扣。

3．导航测试

导航描述用户在一个页面内操作的方式。导航测试可以从以下几个问题进行考虑：导航是否直观？系统的主要部分是否可通过主页存取？是否需要站点地图、搜索引擎或其他的导航帮助？页面结构、菜单、链接的风格是否一致？用户凭直觉是否能知道系统里面的内容在什么地方？

其实导航就是为了让用户知道自己在哪儿，这是什么以及可以去哪儿。那么按以下几个步骤可以测试页面的导航能力。

（1）从网站上随机选择一个页面；

（2）把这个页面打印成黑白的，并把页面头部的浏览器地址栏和下面的版权及公司信息部分去掉；

（3）假装是第一次进入这个网站，并试图回答在哪儿、是什么以及可以去哪儿等问题；

（4）在一张纸上写下你所想的和答案。

这样就可以测试出网站上哪些导航是不合理的。

4．界面测试

用户界面测试（User Interface Testing，又称 UI 测试）指测试软件中的可见外观及其底层与用户交互的部分（菜单、对话框、窗口及其他控件）。

俗话说“人靠衣裳马靠鞍”，良好的外观往往能够吸引眼球，激发顾客（用户）的购买欲望，最终达成商业利益的实现。软件的设计亦如此，Windows XP 在商业上的巨大成功很大一方面来自于它一改往日呆板，以突出“应用”的灰色界面，从“用户体验”角度来设计界面，使界面具有较大的亲和力。就目前的软件设计的发展趋势来说，良好的人机界面设计越来越受到系统分析、设计人员的重视。在软件界面设计强调张扬个性的同时，我们不能忘记软件界面的设计要讲求规矩、简洁、一致、易用，这是一切软件界面设计和测试的必循之道，是软件人机界面在突出自我时的群体定位。美观、规整的软件人机界面破除新用户对软件的生疏感，使老用户更易于上手、充分重用已有使用经验，并尽量少犯错误。由此我们在对软件人机界面进行测试时（设计评审阶段和系统测试阶段结合进行），一定要着重考虑界面的美观度。

用户界面测试是指测试用户界面的风格是否满足客户要求，文字是否正确，页面是否美观，文字、图片组合是否完美，操作是否友好等等。UI 测试的目标是确保用户界面会通过测试对象的功能来为用户提供相应的访问或浏览功能，确保用户界面符合公司或行业的标准，包括用户友好性、人性化、易操作性测试。

下面将界面的一致性测试、信息反馈测试、界面简洁性测试、界面美观度测试、用户动作测试及行业标准测试需要考虑的角度列举出来一些，可供读者在界面测试中参考。

（1）一致性测试

一致性是软件人机界面的一个基本要求。目的是使用户在使用时，能够很快熟悉软件的

操作环境，同时避免对软件操作发生理解歧义。这就要求在进行测试时，需要判断软件的人机界面是否可以作为一个整体而存在。一致性测试需要考虑：

① 提示的格式是否一致；

② 菜单的格式是否一致；

③ 帮助的格式是否一致；

④ 提示、菜单、帮助中的术语是否一致；

⑤ 各个控件之间的对齐方式是否一致；

⑥ 输入界面和输出界面在外观、布局、交互方式上是否一致；

⑦ 命令语言的语法是否一致；

⑧ 功能类似的相关界面在外观、布局、交互方式上是否一致（比如商品代码检索和商品名称检索）；

⑨ 存在同一产品族的时候，与其他产品在外观、布局、交互方式上是否一致（例如 Office 产品族）；

⑩ 同一层次的文字在同一种提示场合（一般情况、凸显、警告等）在文字大小、字体、颜色、对齐方式方面是否一致；

⑪ 多个连续界面依次出现的情况下，界面的外观、操作方式是否一致（当然可能会有例外，比如操作结束的界面）。

（2）信息反馈测试

系统的使用者或许是一个新手，在操作时难免出错，如果界面有友好的输入检查和错误提示功能，那么通过这些信息反馈，用户得到出错提示或任务完成之类的语言，会感觉界面比较亲切而且容易掌握。信息反馈测试时可从以下角度考虑：

① 系统是否接受客户的正确输入并做出提示（例如鼠标焦点跳转）；

② 系统是否拒绝客户的错误输入并做出提示（例如弹出警告框，声响）；

③ 系统显示用户的错误输入的提示是否正确，浅显易懂（例如“ERR004”这样的提示让人不知所云）；

④ 系统是否在用户输入前给出用户具体输入方式的提示（例如网站注册程序）；

⑤ 系统提示所用的图标或图形是否具有代表性和警示性；

⑥ 系统提示用语是否按警告级别和完成程度进行分级（若非某些破坏性操作，请对用户温和一些）；

⑦ 系统在界面（主要是菜单、工具条）上是否提供突显功能（比如鼠标移动到控件时，控件图标变大或颜色变化至与背景有较大反差，当移动开后恢复原状）；

⑧ 系统是否在用户完成操作时给出操作成功的提示（很多系统都缺少这一步，使用户毫无成就感）。

（3）界面简洁性测试

人机界面需要设计得生动而引人入胜，但绝对不是花哨杂乱，所以界面简洁性测试可以从以下角度考虑：

① 用户界面是否存在空白空间（没有空白空间的界面是杂乱无章的，易用性极差）；

② 各个控件之间的间隔是否一致；

③ 各个控件在垂直和水平方向上是否对齐；

④ 菜单深度是否在三层以内（建议不要超出三层，大家可以参考微软的例子）；

⑤ 界面控件分布是否按照功能分组（菜单、工具栏、单选框组、复选框组、Frame 等）；

⑥ 界面控件本身是否需要通过滑动条的滑动来显示数据（建议采用分页显示并提供数据排序显示功能）；

实际上，一个处理该类测试的原则性的东西就是：清除多余的东西，尽可能分组。

（4）界面美观度测试

如果界面美观漂亮，能给人以美的享受，那么用户就乐于访问你的系统，客户更容易同意你的方案。“美是对比的产物”，界面美观度测试从以下角度考虑：

① 前景与背景色搭配是否反差过大；

② 前景与背景色是否采用较为清淡的色调而不是深色（比如用天蓝色而不用深蓝色和墨绿色）；

③ 系统界面是否采用了超过三种的基本色（一般情况下不要超过三种）；

④ 字体大小是否与界面的大小比例协调（一般中文采用宋体 9-12，英文采用 Arial 或 Times New Roman，日文采用 SimSun 或明朝）；

⑤ 按钮较多的界面是否禁止缩放（一般情况下不宜缩放，最好禁止最大、最小化按钮）；

⑥ 系统是否提供用户界面风格自定义功能，满足用户个人偏好；

（5）用户动作测试

用户不会有太多的耐心对待他们期待的系统，如果软件系统能让用户尽量少动手少记命令，那么用户就会喜欢使用它。用户动作性测试可以从以下角度判断是否能让用户“偷懒”：

① 是否存在用户频繁操作的快捷键；

② 是否允许动作的可逆性（Undo，Redo）；

③ 界面是否有对用户的记忆要求；

④ 系统的反应速度是否符合用户的期望值；

⑤ 是否存在更便捷、直观的方式来取代当前的界面的显示方式（例如用菜单界面代替命令语言界面）；

⑥ 用户在使用时任何时候是否能开启帮助文档（F1）；

⑦ 系统是否提供模糊查询机制和关键字提示机制减少用户的记忆负担（例如清华紫光输入法的模糊音设定）；

⑧ 是否对可能造成长时间等待的操作提供操作取消功能；

⑨ 是否支持对错误操作进行可逆性处理，返回原有状态；

⑩ 是否采用相关控件（例如日历，计算器等）替代用户手工键盘输入；

⑪ 选项过多的情况下是否采用下拉列表或者关键字检索的方式供用户选择；

⑫ 系统出错是否存在恢复机制使用户返回出错前状态（例如 Office XP 的文件恢复）；

⑬ 在用户输入数据之前，用户输入数据后才能执行的操作是否被禁止（例如特定的按钮变灰）；

⑭ 系统是否提供“所见即所得”或“下一步提示”的功能（例如预览）。

（6）行业标准测试

“国有国法，行有行规”，设计系统界面也要注意使用系统的客户所在行业的标识体系，测试人员对这些行业符号体系必须有所了解，行业标准测试可以从以下一些角度来考虑是否与行业标识体系冲突：

① 界面使用的图符、声音是否符合软件所面向领域的行业符号体系标准；

② 界面使用的术语是否符合软件所面向领域的行业命名标准；

③ 界面的颜色是否与行业代表色彩较为相近；

④ 界面的背景是否能够反映行业相关主题（比如：反映环保的背景一般采用自然风光作为背景）；

⑤ 界面的设计是否反映行业最新的理念和大众趋势；

除此之外，还要考虑软件开发商自身的个性或徽标，而开发商既要突显自身的特性又不能喧宾夺主，对此可从以下角度出发：

① 软件的安装界面是否有单位介绍或产品介绍，并拥有自己的图标；

② 软件的安装界面是否在界面上不同于通用的安装工具生成的界面（例如金山快译的安装界面就比较有特色）；

③ 主界面的图标是否为制作商的图标；

④ 系统启动需要长时间等待时，是否存在 Splash 界面，它是否包含或反映制作者信息；

⑤ 软件是否有版本查看机制，版本说明上是否有制作者或是用户的标识；

⑥ 软件的界面的色彩、背景、布置是否与同类产品有不同之处，如果有，是否更为简洁、美观；

⑦ 软件界面操作与同类产品相比，是否能够减少用户输入的频繁度；

⑧ 软件界面操作与同类产品相比，是否在出错预防机制和提示上更为直观、醒目；

⑨ 软件界面是否为特殊群体或是特殊的应用提供相应的操作机制（例如 Windows 系统的放大镜）。

5．兼容性测试

兼容性就是指协调性。通俗地讲兼容性分为硬件兼容性和软件兼容性两个方面：

（1）硬件上就是指计算机的各个部件，比如 CPU、显卡或主板等组装到一起以后的情况，会不会相互有影响，能不能很好的协同运作。

（2）软件上就是指计算机的软件之间能否很好的协作，互相之间会不会有影响？还有软件和硬件之间能否很好地提高工作效率，会不会互相影响导致系统的崩溃。

兼容性测试包括平台测试、浏览器测试、分辨率测试、连接速度测试和组合测试等。

（1）目前市场上有很多不同种类的操作系统，比如常见的有 Windows、Linux 及 UNIX 等。那么 Web 程序在这些操作系统上能否正常运行，发布之前就要进行平台的兼容性测试。

（2）浏览器是 Web 应用系统客户端使用的最主要的构件，市场上浏览器的种类也极其繁多，它们的规格也不尽相同。比如框架和层次风格在不同的浏览器中就有不同的显示或根本不显示，所以发布 Web 应用系统之前也要进行浏览器的兼容性测试。

（3）客户端用户使用的显示分辨率有多种多样，常见的有 800×600、1024×768、1280×800、1024×640、1440×900 等，那么这些不同分辨率模式下页面能否正常显示？Web 应用程序就有必要进行分辨率的兼容性测试，以保证不同分辨率下页面可以正常运行。

（4）客户端用户在浏览网站或下载资源的时候，如果等待时间比较长则不会耐心等待，可能会转到其他网站去。所以对连接速率的测试也是很重要的。测试速度的时候测试人员可能使用的带宽比较宽，而客户却使用较窄带宽，这时就需要对连接速度的兼容性问题进行测试，判断是否在较窄带宽下 Web 应用系统也能在用户可以忍受的范围内运行。

（5）组合测试就是将以上几种兼容性测试组合起来进行，比如在不同平台的不同浏览器下运行，或者兼具不同的分辨率与带宽。这样也是为了使测试内容更全面，更多地发现系统

漏洞。

6．文档测试

文档测试（Documentation testing）是一个很重要的阶段，也可以叫评审。就是阅读需求分析文档、概要设计文档及详细设计文档等，找出错误的或者不合适的地方。软件测试技术自出现并发展到现在，不仅仅只是在软件开发结束后再测试，而是深入到软件开发的各个阶段中，从需求分析就要进行测试，直到软件开发完毕，最后到实施维护。

文档测试关注于文档的正确性，文档有三大类，分别是开发文件、用户文件和管理文件。

（1）开发文件包括：

- 可行性研究报告
- 软件需求说明书
- 数据要求说明书
- 概要设计说明书
- 详细设计说明书目
- 数据库设计说明书
- 模块开发卷宗

（2）用户文件包括：

- 用户手册
- 操作手册

（3）管理文件包括：

- 项目开发计划
- 测试计划
- 测试分析报告
- 开发进度月报
- 项目开发总结报告

软件测试中的文档测试主要是对相关的设计报告和用户使用说明进行测试，对于设计报告主要是测试程序与设计报告中的设计思想是否一致；对于用户使用说明进行测试时，主要是测试用户使用说明书中对程序操作方法的描述是否正确，重点是用户使用说明中提到的操作例子要进行测试，保证采用的例子能够在程序中正确完成操作。

这里重点介绍一下帮助文档测试。帮助文档应该提供所有规格说明和各种操作命令用法，可以帮助和引导用户在使用软件遇到困难时自己寻找解决办法。

对帮助文档的测试可以从以下几个方面着手：

（1）前后一致性；

（2）内容完整性；

（3）可理解性；

（4）方便性。

一个好的系统应该提供一套完整而详细的帮助文档，使系统尽善尽美。

到此为止我们已经介绍了软件测试所使用的大部分测试策略和设计用例的方法，值得一提的是，通常在按照所有测试用例测试完毕，测试中发现的问题全部解决后，还要进行下一轮的测试，即回归测试。

进行文档测试时，应该先列出项目过程中的文档清单，然后对各文档进行测试总结，完

成表 2-11 的检查表。

表 2-11　文档测试结果

文档（版本/日期）	已创建或可用	已被接收或已经过复审	作者来源	备　　注
可行性分析报告	是□　　否□	是□　　否□		
软件需求定义	是□　　否□	是□　　否□		
软件系统分析（STD，DFD，CFD，DD）	是□　　否□	是□　　否□		
软件概要设计	是□　　否□	是□　　否□		
软件详细设计	是□　　否□	是□　　否□		
软件测试需求	是□　　否□	是□　　否□		
硬件可行性分析报告	是□　　否□	是□　　否□		
硬件需求定义	是□　　否□	是□　　否□		
硬件概要设计	是□　　否□	是□　　否□		
硬件原理图设计	是□　　否□	是□　　否□		
硬件结构设计（包含 PCB）	是□　　否□	是□　　否□		
FPGA 设计	是□　　否□	是□　　否□		
硬件测试需求	是□　　否□	是□　　否□		
PCB 设计	是□　　否□	是□　　否□		
USB 驱动设计	是□　　否□	是□　　否□		
Tunner BSP 设计	是□　　否□	是□　　否□		
MCU 设计	是□　　否□	是□　　否□		
模块开发手册	是□　　否□	是□　　否□		
测试时间表及人员安排	是□　　否□	是□　　否□		
测试计划	是□　　否□	是□　　否□		
测试方案	是□　　否□	是□　　否□		
测试报告	是□　　否□	是□　　否□		
测试分析报告	是□　　否□	是□　　否□		
用户操作手册	是□　　否□	是□　　否□		
安装指南	是□　　否□	是□　　否□		

7．本地化测试

本地化测试（Localization testing），本地化测试的对象是软件的本地化版本。本地化测试的目的是测试特定目标区域设置的软件本地化质量。本地化测试的环境是在本地化的操作系统上安装本地化的软件。从测试方法上可以分为基本功能测试，安装/卸载测试，当地区域的软硬件兼容性测试。测试的内容主要包括软件本地化后的界面布局和软件翻译的语言质量，包含软件、文档和联机帮助等部分。

本地化就是翻译产品的 UI，有时也更改某些初始设置以使产品适合于另一个地区。本地化测试检查针对特定目标区域或区域设置的产品本地化质量。此测试基于全球化测试的结果，后者验证对特定区域或区域设置的功能性支持。本地化测试只能在产品的本地化版本上进行，不对本地化质量进行测试。

软件本地化测试是软件本地化项目的一个重要组成部分，是提高软件本地化质量的重要手段，是控制软件本地化质量的关键措施。软件本地化测试的目的是为了发现本地化的软件中的错误和缺陷，通过修复这些错误和缺陷，提高软件本地化质量。更详细的定义可以描述为，软件本地化测试是根据软件本地化各阶段的测试计划和规格说明，精心设计一批测试用例（即输入数据及其预期的输出结果），并利用这些测试用例去运行本地化软件，以发现程序错误和缺陷的过程。综合的软件本地化测试解决方案，可以保证软件发布进度、降低支持和维护成本，并保证产品有上乘的质量。

软件本地化测试是一个系统工程，包含多个紧密联系的环节和内容。软件本地化测试作为保证软件本地化质量和可靠性的技术手段，随着软件国际市场的激烈竞争和软件用户对质量要求的不断提高，软件本地化测试在软件本地化项目中的作用更加突出。软件本地化测试的关键在于软件供应商（Software Provider）和本地化提供商（Localization Vendor）对测试的高度重视，包括测试资源、测试文档、测试流程、测试方法和测试管理等方面有效准备和正确实施。

随着跨国公司产品国际化战略的深入发展，产品本地化需求快速增长。根据 LISA 提供的 2003 年的数据，在 GILT 服务领域，大型国际公司产品的翻译外包比例达到 78%，产品的桌面排版（DTP）占到 64%，产品本地化达 27%，产品的质量保证（QA）占 13%，详细数据比例如图 2-29 所示。所以软件本地化测试也越来越受到更多的重视。

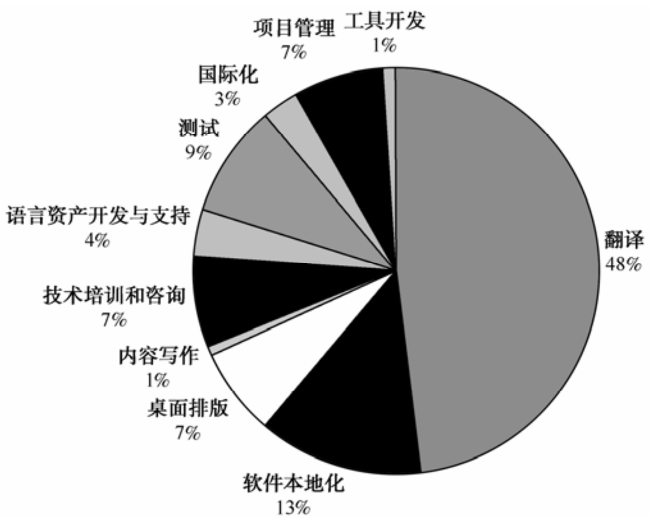


图 2-29　大型国际公司产品的业务外包比例

（1）软件本地化工程

软件本地化工程是对本地化的文件进行资源文件抽取、格式转换、本地化编译和修正缺陷的过程。它融合了软件工程、翻译技术和桌面出版等技术，是软件本地化不可缺少的环节。

经过软件本地化工程处理后，可以提高其他软件本地化工作（翻译、桌面排版、测试、项目管理）的工作效率，并且有助于保持本地化内容的一致性。

软件本地化工程包括软件、联机帮助和图像的本地化工程，分别对软件程序、软件的联机帮助和本地化软件的图像进行格式转化、内容本地化、重新编译和修正缺陷等处理。

对于包含软件本地化翻译、工程处理、测试和桌面排版的大型软件本地化项目的流程如

图 2-30 所示。

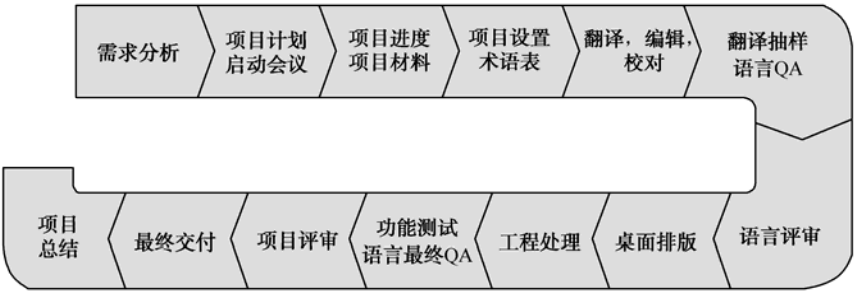


图 2-30　软件本地化项目的处理流程

（2）本地化测试包括的内容

① 本地化测试过程中的测试工作集中在以下两个方面，进行本地化测试时可以着重从这两个方面进行考虑：

- 受本地化影响的方面，如 UI 和内容；
- 区域性或区域设置特定的、语言特定的和地区特定的方面。

② 另外，本地化测试还应包括：

- 基本功能测试；
- 在本地化环境中运行的安装和升级测试；
- 根据产品的目标地区计划应用程序和硬件兼容性测试。可以选择 Windows 2000 的任何语言版本作为测试平台。然而，必须安装目标语言支持。

③ 用户界面和语言的本地化测试应包括的项有：

- 验证所有应用程序资源；
- 验证语言的准确性和资源属性；
- 版式错误；
- 书面文档、联机帮助、消息、界面资源、命令键顺序等的一致性检查；
- 确认是否遵守系统、输入和显示环境标准；
- 用户界面可用性；
- 评估文化适合性；
- 检查政治上敏感的内容。

④ 当交付本地化产品时，确保包含本地化文档（手册、联机帮助、上下文帮助等）。要检查的项包括：

- 翻译的质量；
- 翻译的完整性；
- 所有文档和应用程序 UI 中使用的术语一致。

（3）发现缺陷的基本方法

对于软件本地化测试，本地化提供商主要进行外观测试和本地化语言测试，软件供应商主要进行国际化测试和功能测试。软件在本地化之前，必须先经过软件国际化测试和可本地化性能测试等功能测试，然后再编译本地化版本。软件本地化提供商应该重点处理本地化提供商可以解决和擅长的调整本地化软件用户界面布局和语言翻译等问题，对于测试过程报告的软件硬编码（指直接嵌入在代码中的需要本地化的字符）和软件自身的功能错误只能由软

件提供商处理。由于软件本地化和源语言软件开发一起进行，因此，软件本地化测试通常要测试多个功能不断完善和丰富的本地化版本。这些不同的本地化版本测试的重点和具体测试内容各不相同。本地化软件缺陷，具有比较鲜明的特征，发现本地化软件缺陷具有内在规律。

① 按照一定的顺序排查软件缺陷

本地化软件缺陷可以分为用户界面错误、语言质量错误和功能错误等。这些不同类型的错误有时同时出现在软件的某个部分，为了更全面地发现这些缺陷需要遵循一定的测试排查步骤。以测试一个本地化文本框为例。

a．首先查看文本框控件的用户界面错误

● 是否有被截断（Truncation）的错误；

● 是否控件布局不整齐或者重叠；

● 是否丢失热键；

● 与英文软件的热键不一致；

● 文本字体类型和大小错误；

● 字符显示乱码错误。

b．其次，查看对话框语言质量错误

● 是否有遗漏的需要本地化的英文文字；

● 本地化的文字是否内容正确、专业和流畅；

● 是否存在多余的翻译，例如，对不该翻译的内容进行了翻译；

● 是否翻译的内容带有敏感的政治问题和与目标市场的风俗传统不一致；

● 标点符号是否符合本地化语言用户的使用习惯。

c．再次，查看对话框功能错误

● 每个要测试的按钮功能是否起作用；

● 每个要测试的按钮功能是否正确；

● Tab 键的跳转是否正确；

● 控件的热键是否起作用。

按照以上的顺序测试，主要是为了按照从易到难的原则，全面地查找软件缺陷，而不是说功能错误是不重要的，相反功能错误是优先级比较高的错误，应该正确处理。

② 对照源语言软件确认缺陷

绝大多数本地化软件缺陷都是由于本地化过程引起的，但是也有少量错误是由于源语言软件不正确的设计引起的。这类错误表现为同一个错误在源语言软件和本地化版本中都可以复现，例如某些功能不起作用，热键重复等。还有一种错误只在本地化软件中复现，但是本地化工程师无法解决，需要软件开发人员修改编码进行修正，例如不支持双字节的输入、显示、输出等，这些属于源语言软件的国际化设计错误。

因此，如果发现了功能错误和热键等软件缺陷，应该在相同测试环境的源语言软件上，进行同样步骤的测试，将测试结果进行比较，并且报告中应该指出该软件缺陷是否也存在于源语言软件上。

③ 利用软件缺陷的“扎堆”现象

软件缺陷的“扎堆”现象就是被测软件的某些部分往往出现不止一个错误。因此，如果在某一部分发现了很多错误，应该进一步仔细测试是否还包含了更多的软件缺陷。

软件缺陷的“扎堆”现象的常见形式：

- 对话框的某个控件功能不起作用，可能其他控件的功能也不起作用。
- 某个文本框不能正确显示双字节字符，则其他文本框也可能不支持双字节字符。
- 联机帮助某段文字的翻译包含了很多错误，与其相邻的上下段的文字可能也包含很多的语言质量问题。
- 安装文件某个对话框的“上一步”或“下一步”按钮被截断，则这两个按钮在其他对话框中也可能被截断。

软件测试中重视软件缺陷的“扎堆”现象，有助于发现更多的软件缺陷。

④ 关注测试容易产生软件缺陷的部分

软件缺陷几乎无处不在，可以出现在软件的任何地方。但是，软件缺陷的产生也有些规律，软件的某些部分更容易产生软件缺陷，这些部分是软件错误的“重灾区”，在测试期间应该引起重视和测试。

容易产生软件本地化缺陷的软件部分包括：

- 软件的“关于”对话框中容易产生版权（C）和商标（TM）等字符的显示错误。
- 软件中与语言设置相关的部分容易产生软件国际化错误，例如，排序方式、数字格式、日期格式、时间格式、货币符号、度量衡、电话号码格式、纸张类型等。
- 对话框和菜单中容易产生热键错误。对话框中容易产生用户界面错误。
- 终端用户许可协议（EULA）容易产生国家和地区名称翻译的政治敏感错误，例如，将地区翻译成国家等。
- 联机帮助文件中容易产生与软件界面不一致的术语翻译错误。

⑤ 参考其他语言版本测试发现的缺陷

如果测试项目组同时在测试不同语言的相同本地化版本，应该尽量参考其他本地化软件测试过程中发现的缺陷，例如，在报告的缺陷管理库中搜索报告的缺陷，与其他语言的测试工程师交流等，查看自己是否漏测、漏报了某些软件缺陷，并且努力补上。另外，有些本地化错误可能仅仅出现在某个本地化语言版本中，属于该语言版本的本地化过程引起的错误，在报告软件缺陷时，说明本缺陷不出现在其他本地化语言版本上，将有助于软件工程师正确隔离软件缺陷，分配给合适的人员处理缺陷。

⑥ 使用测试辅助工具

根据测试的不同内容，选择合适的测试辅助工具，可以有效地发现软件缺陷。例如，使用文件和文件夹比较工具（例如，Beyond Compare 或 WinDiff）可以方便地比较英文和本地化 build 的文件和文件夹内容。Html QA 可以很方便、准确地定位编译的本地化联机帮助文件的链接是否完整、正确。

本地化软件的缺陷的类型具有明显的特征，测试工程师只要熟悉这些特征，按照适当的测试顺序，仔细地观察比较可以发现绝大多数软件缺陷。另外，本地化测试也是不断实践、不断丰富测试经验的过程，测试人员之间加强交流，不断自我总结测试经验，可以不断提高寻找和发现软件缺陷的能力。

2.5.8 拓展任务

独立完成天天超市管理系统的其他测试。

第 3 章　测 试 执 行

本章介绍利用测试工具 LoadRunner 8.0 进行测试脚本的录制，脚本的调试与完善，测试执行，测试结果分析。通过本章的讲解，能够完整地浏览测试执行的全过程。

本章重点：

- 测试执行过程中参数的设置
- 性能测试的执行过程

当测试计划、测试用例都完成时，我们就要开始执行测试了。

工作任务 3.1　测试执行概述

3.1.1　测试执行概述

1．工作任务描述

当测试计划、测试用例都完成时，我们就要开始执行测试了。

在测试用例执行过程中，搭建测试环境是第一步。一般来说，软件产品提交测试后，开发人员应该提交一份产品安装指导书，在指导书中详细指明软件产品运行的软 硬件环境，此外，应该给出被测试软件产品的详细安装指导书，包括安装的操作步骤、相关配置文件的配置方法等。对于复杂的软件产品，尤其是软件项目，如果没有安装指导书作为参考，在搭建测试环境过程中会遇到种种问题。

如果开发人员拒绝提供相关的安装指导书，搭建测试中遇到问题的时候，测试人员可以要求开发人员协助，这时候，一定要把开发人员解决问题的方法记录下来，避免同样的问题再次请教开发人员，这样会招致开发人员的反感，也降低了开发人员对测试人员的认可程度。

2．工作过程

（1）全方位地观察测试用例执行结果

测试执行过程中，当测试的实际输出结果与测试用例中的预期输出结果一致的时候，是否可以认为测试用例执行成功了？答案是否定的，即便实际测试结果与测试的预期结果一致，也要查看软件产品的操作日志、系统运行日志和系统资源使用情况，来判断测试用例是否执行成功了。全方位观察软件产品的输出可以发现很多隐蔽的问题。以前，我在测试嵌入式系统软件的时候，执行某测试用例后，测试用例的实际输出与预期输出完全一致，不过在查询 CPU 占用率的时候，发现 CPU 占用率高达 90 %，后来经过分析，软件运行的时候启动了若干个 1ms 的定时器，大量地消耗了 CPU 资源，后来通过把定时器调整到 10ms，CPU 的占用率降为 7%。如果观察点单一，这个严重消耗资源的问题就无从发现了。

（2）加强测试过程记录

测试执行过程中，一定要加强测试过程记录。如果测试执行步骤与测试用例中描述的有差异，一定要记录下来，作为日后更新测试用例的依据；如果软件产品提供了日志功能，比

如有软件运行日志、用户操作日志，一定在每个测试用例执行后记录相关的日志文件，作为测试过程记录，一旦日后发现问题，开发人员可以通过这些测试记录方便的定位问题，而不用测试人员重新搭建测试环境，为开发人员重现问题。

（3）及时确认发现的问题

测试执行过程中，如果确认发现了软件的缺陷，那么可以毫不犹豫地提交问题报告单。如果发现了可疑问题，又无法定位是否为软件缺陷，那么一定要保留现场，然后通知相关开发人员到现场定位问题。如果开发人员在短时间内可以确认是否为软件缺陷，测试人员给予配合;如果开发人员定位问题需要花费很长的时间，测试人员千万不要因此耽误自己宝贵的测试执行时间，可以让开发人员记录重现问题的测试环境配置，然后，回到自己的开发环境上重现问题，继续定位问题。

（4）提交缺陷时与开发的关系处理

测试执行过程中，当你提交了问题报告单，可能被开发人员无情驳回，拒绝修改。这时候，只能对开发人员晓之以理，做到有理、有据，有说服力。首先，要定义软件缺陷的标准原则，这个原则应该是开发人员和测试人员都认可的，如果没有共同认可的原则，那么开发人员与测试人员对问题的争执就不可避免了。此外，测试人员打算说服开发人员之前，考虑是否能够先说服自己，在保证可以说服自己的前提下，再开始与开发人员交流。

（5）及时更新测试用例

测试执行过程中，应该注意及时更新测试用例。往往在测试执行过程中，才发现遗漏了一些测试用例，这时候应该及时的补充；往往也会发现有些测试用例在具体的执行过程中根本无法操作，这时候应该删除这部分用例；也会发现若干个冗余的测试用例完全可以由某一个测试用例替代，那么删除冗余的测试用例。

总之，测试执行的过程中及时地更新测试用例是很好的习惯。不要打算在测试执行结束后，统一更新测试用例，如果这样，往往会遗漏很多本应该更新的测试用例。

（6）提交一份优秀的问题报告单

软件测试提交的问题报告单和测试日报一样，都是软件测试人员的工作输出，是测试人员绩效的集中体现。因此，提交一份优秀的问题报告单是很重要的。缺陷报告单中最关键的几个部分：第一部分是发现缺陷的环境，包括软件环境、硬件环境等；第二部分是缺陷的基本描述；第三部分是开发人员对缺陷的解决方法。通过对上述缺陷报告单的三个部分进行仔细分析，从中掌握了软件产品最常见的基本问题，并吸收了其他软件测试人员的工作经验。

最关键的域就是“问题描述”，这是开发人员重现问题，定位问题的依据。问题描述应该包括以下几部分内容：软件配置、硬件配置、测试用例输入、操作步骤、输出、当时输出设备的相关输出信息和相关的日志等。

软件配置：包括操作系统类型版本和补丁版本、当前被测试软件的版本和补丁版本、相关支撑软件，比如数据库软件的版本和补丁版本等。

硬件配置：计算机的配置情况，主要包括CPU、内存和硬盘的相关参数，其他硬件参数根据测试用例的实际情况添加。如果测试中使用网络，那么包括网络的组网情况，网络的容量、流量等情况。硬件配置情况与被测试产品类型密切相关，需要根据当时的情况，准确翔实地记录硬件配置情况。

测试用例输入、操作步骤、输出：这部分内容可以根据测试用例的描述和测试用例的实际执行情况如实填写。

输出设备的相关输出信息：输出设备包括计算机 显示器、打印机、磁带等输出设备，如果是显示器可以采用抓屏的方式获取当时的截图也可以录制视频，其他的输出设备可以采用其他方法获取相关的输出，在问题报告单中提供描述。

日志信息：规范的软件产品都会提供软件的运行日志和用户、管理员的操作日志，测试人员应该把测试用例执行后的软件产品运行日志和操作日志作为附件，提交到问题报告单中。

3.1.2　应知应会

测试用例执行结果的表示方法：

FastPoint 在一个帖子中把执行测试用例后的结果归纳为三种情况：

（1）Pass：表示执行测试用例后，没有发现 Bug；

（2）Fail：表示执行测试用例后，发现了软件 Bug；

（3）Errror：表示执行测试用例过程中，发生了错误，没有成功执行该测试用例。

在这里补充一下我在测试工作中如何处理测试用例的执行结果的：

我们习惯上把测试用例的执行结果分得更详细一点：

（1）Pass：表示执行测试用例后，没有发现 Bug；

（2）Fail：表示执行测试用例后，发现了软件 Bug；

（3）Block：表示该测试用例无法执行，可能是测试用例描述不完整，或者步骤与被测软件不符合；

（4）Skip：表示该测试用例不需要在当前测试阶段执行；

（5）Issue：表示该测试用例虽然可以执行，但是输出结果无法判断是否正确，需要请有关人员确认是否是 Bug 或者是测试用例本身的设计问题。

3.1.3　拓展任务

分析天天超市管理系统在测试执行时应注意的问题。

工作任务 3.2　测试执行准备工作

3.2.1　对执行测试人员的培训

1．工作任务描述

在测试执行过程中，参与测试执行的人员不一定是一开始就参与了需求分析和功能分析、计划，以及测试用例的设计，因为测试用例一般是有经验的测试工程师设计的，而测试执行对参与人员的要求相对低一些，所以后加入到该项目的人员对项目背景不是很了解，对该项目的测试计划、测试用例设计也不是很清楚，所以要对这些测试人员进行相关的测试培训和产品说明。对测试执行人员的培训主要包括以下几点。

2．工作过程

（1）项目介绍

对测试执行人员介绍本项目的背景，及客户的基本要求，提供产品说明书、需求文档。

（2）对该项目测试计划、测试用例的介绍

根据用户的要求与需求文档，设计测试计划书和测试用例，让后参与的人员了解该项目

的测试计划和测试用例的设计，这是执行测试的主要依据。

（3）对相关知识的介绍

本项目所采用的技术与测试工具要有一个系统的说明，参与执行测试的人员必须熟悉相关的技术与测试工具，为即将开始执行的测试做好准备。

（4）测试团队的建设。

3.2.2 测试任务及进度的安排

1．工作任务描述

根据具体的项目合理安排工作量，合理分配人员及设置时间节点，如表 3-1 所示。

2．工作过程

表 3-1 测试任务及进度安排

测试阶段	测试任务	工作量估计	人员分配	起止时间
第一阶段	功能测试	8 课时	每名教师带领 10 人	
第二阶段	界面测试	4 课时	1 名教师带领全部学生	
第三阶段	链接测试	4 课时	1 名教师带领全部学生	
第四阶段	兼容性测试	4 课时	每名教师带领 10 人	
第五阶段	性能测试	4 课时	每名教师带领 10 人	

3.2.3 自动化测试的执行

1．工作任务描述

本节在讲自动化测试执行过程中，采用的教学项目是 LoadRunner 自带的网上订票系统为例进行讲解的，如表 3-2 所示。

2．工作过程

（1）功能测试用例执行过程

表 3-2 测试用例

<table>
<tr><td>用例编号</td><td>UserCase_01</td><td>模块名称</td><td colspan="3">用 户 注 册</td></tr>
<tr><td>用例作者</td><td></td><td>设计日期</td><td></td><td>测试签字</td><td></td></tr>
<tr><td>审核结果</td><td></td><td>审核时间</td><td></td><td>审核签字</td><td></td></tr>
<tr><td>测试类型</td><td colspan="5">功能性自动化测试</td></tr>
<tr><td>权限</td><td colspan="5">用户</td></tr>
<tr><td>角色</td><td colspan="5">普通用户</td></tr>
</table>

用例描述	通过 IE 浏览器，访问网上订票系统登录网址，输入正确的用户名、密码以及密码确认等注册信息，进行用户注册		
前置条件	网上购物系统应用服务器部署正确，网络畅通，能够通过浏览器正常访问		
步　骤	在测试终端机，运行测试工具软件 LoadRunner； 载入测试用例 UserCase_02 对应的测试脚本； 运行测试脚本，来回放模拟的测试操作； 记录并分析测试结果		
约束条件	用户名称不能重复		
测试脚本			
预期结果	测试脚本正常运行，没有出现异常； 测试工具成功模拟用户操作，通过 IE 浏览器访问网上订票系统注册网址； 在用户注册界面测试工具将模拟操作者填入用户名、密码及密码确认信息； 单击［确定］按钮，注册成功完成		
实际结果	测试脚本被执行，无异常，成功完成了模拟网上订票系统的用户注册操作		
结　论	□通过　　□未通过	测试日期	

操作步骤：

① 测试脚本的录制

a．启动 Visual User Generator 后，在主窗口，单击“File”菜单，选择“New…”菜单项来新建一个测试脚本。如图 3-1 所示。

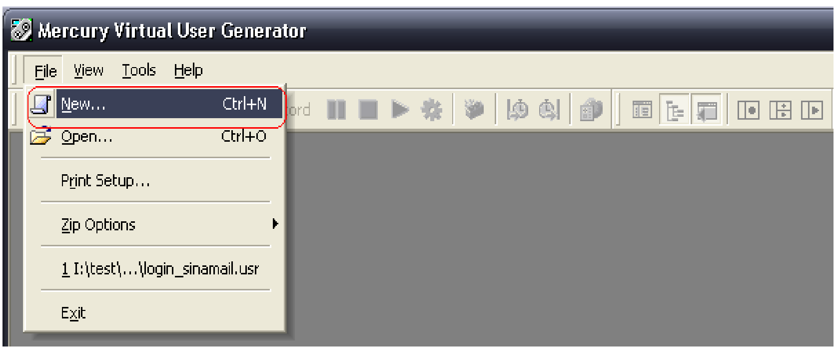


图 3-1　新建脚本界面

b．新建测试脚本第一步是选择系统通信的协议，在“New Virtual User”对话框中单击左侧［New Single Protocol Script］按钮，来选择系统通信协议，由于我们练习实例《网上订票系统》属于 B/S 结构的系统，因此选择“Web（HTTP/HTML）”协议，如图 3-2 所示。

c．在“New Virtual User”对话框单击［OK］按钮，将进入“Start Recording”对话框，如图 3-3。

d．以LoadRunner自带的网上订票系统为例，首先，需要在“URL”地址框中输入 http://localhost:1080/mercuryWebTours，即网上订票系统的用户注册主页地址，然后输入“Record Into Action”，即存放操作脚本的函数名称，默认函数名称为Action，也可以单击［New］

按钮，输入新的函数名称，以上两项设置完成后单击［OK］按钮，将弹出“Recording…”对话框，如图 3-4 所示，便开始录制测试脚本。

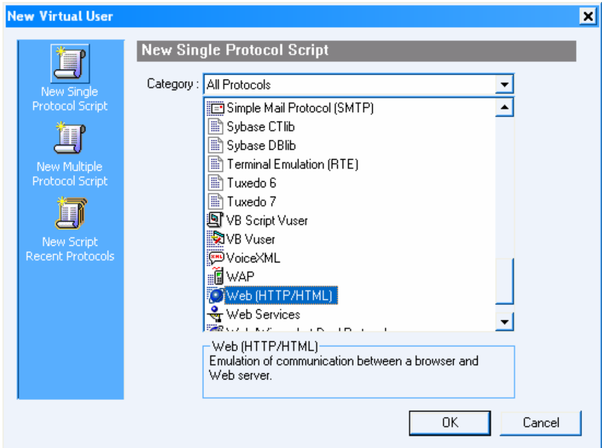


图 3-2　选择协议界面

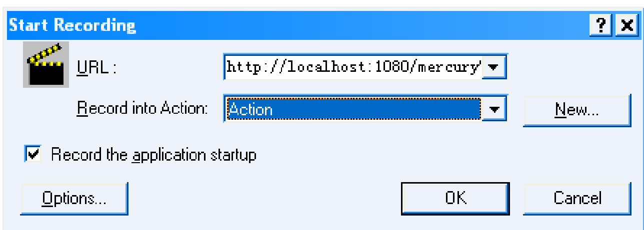


图 3-3　录制配置界面

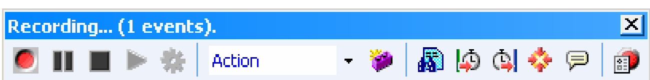


图 3-4　录制界面

e．接下来，就可以根据用例 UserCase_01 中的操作步骤来操作被测试的网上订票系统，首先，测试工具会自动打开 IE 浏览器，并登录到网上订票系统的用户注册页面，如图 3-5 所示。

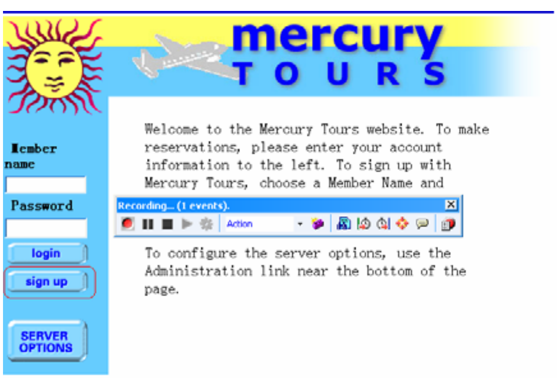


图 3-5　网上订票系统注册页面

f．在用户注册页面单击［sign up］按钮，将进入注册信息输入页面，如图 3-6 所示。

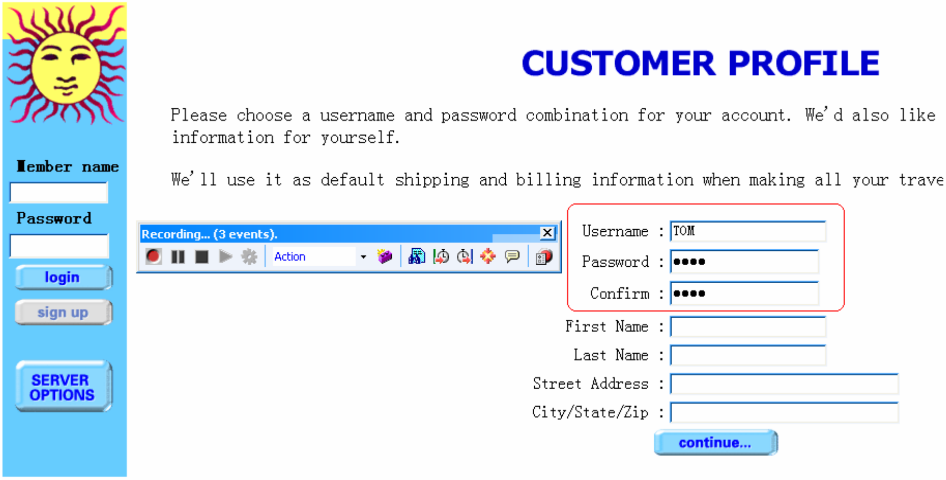


图 3-6　注册信息输入页面

g．输入用户名称、密码及密码确认等信息后，单击［continue…］按钮，将进入用户注册成功的信息页面，如图 3-7 所示。



图 3-7　用户注册成功的信息页面

到此，测试用例 case1 的操作步骤执行完成，单击“Recording…”对话框中的［Stop］按钮，来结束测试脚本的录制。

② 测试脚本调试

在网上订票系统的用户注册功能中，用户名称存在唯一性约束，利用测试工具录制基本测试脚本后，重复执行脚本时，会因为用户名称重复而失败，并返回如图 3-8 所示的测试结果。

解决此问题，可以通过将用户名称使用参数替代方法来实现，具体操作如下（输入参数化的详细情况参见第 5 章）。

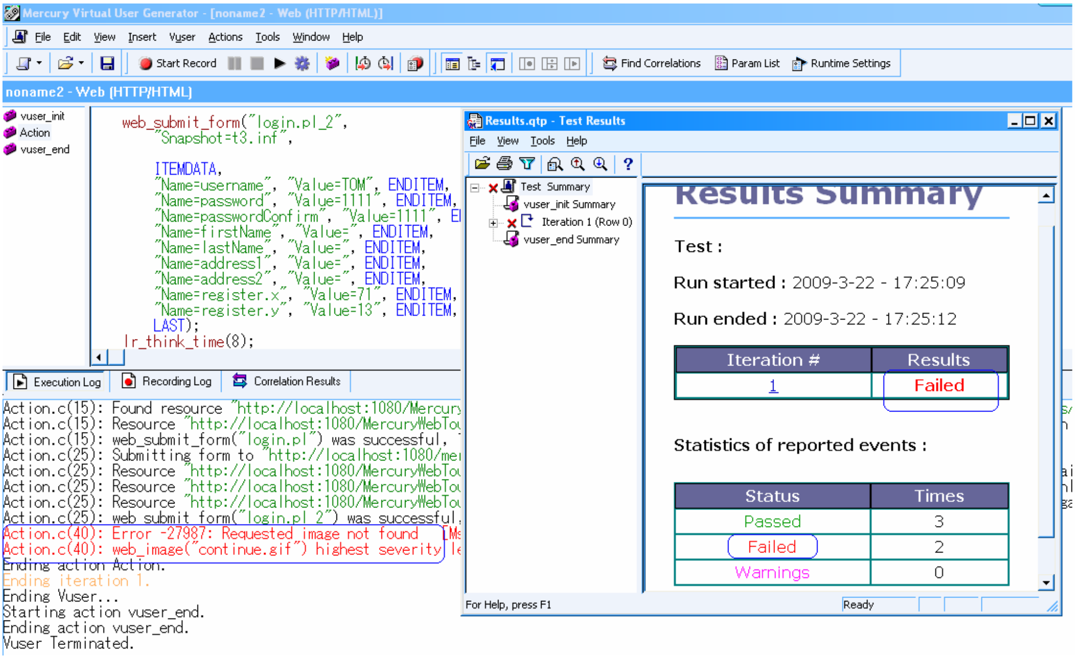


图 3-8　测试结果

a．在 View Script 视图中设置脚本参数比较方便，如图 3-9 所示，在主窗口中单击菜单“View” -> “Script View”菜单项或［View Script］按钮，切换到 View Script 视图。

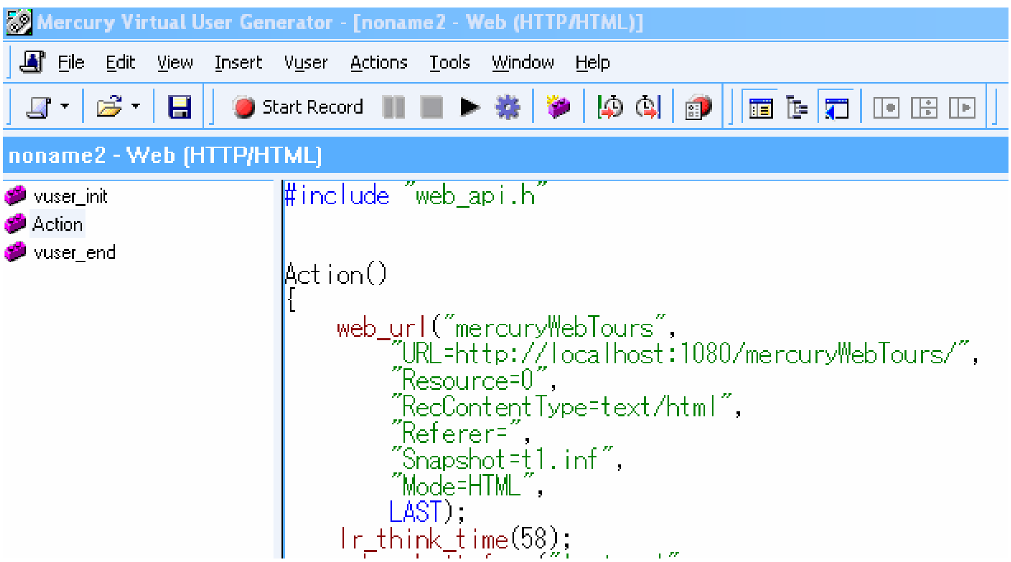


图 3-9　Action 主窗口

b．如图 3-10 所示，在如下代码中，"Name=username", "Value=TOM"，使用鼠标选中要替换的常量值“TOM”，单击鼠标右键，将弹出菜单，选择菜单项“Replace with a parameter”，将弹出如图 3-11 所示的对话框，输入参数名称及参数类型等信息。

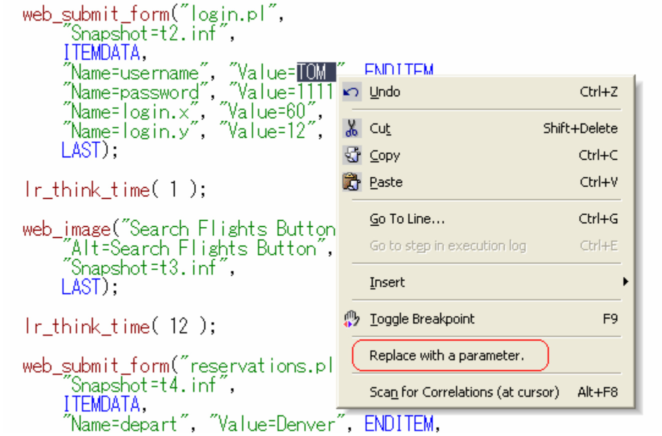


图 3-10　参数递带菜单

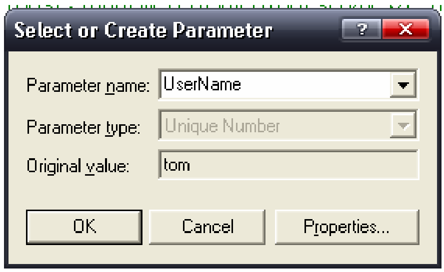


图 3-11　输入参数信息窗口

在创建参数对话框中，输入参数名称（如：UserName），参数类型选择“Unique Number”，这样用唯一的数值来替代代码中的常量，便可解决用户名称不能重复的约束。

c．单击［Properties］按钮，进入属性设置界面，如图 3-12 所示。

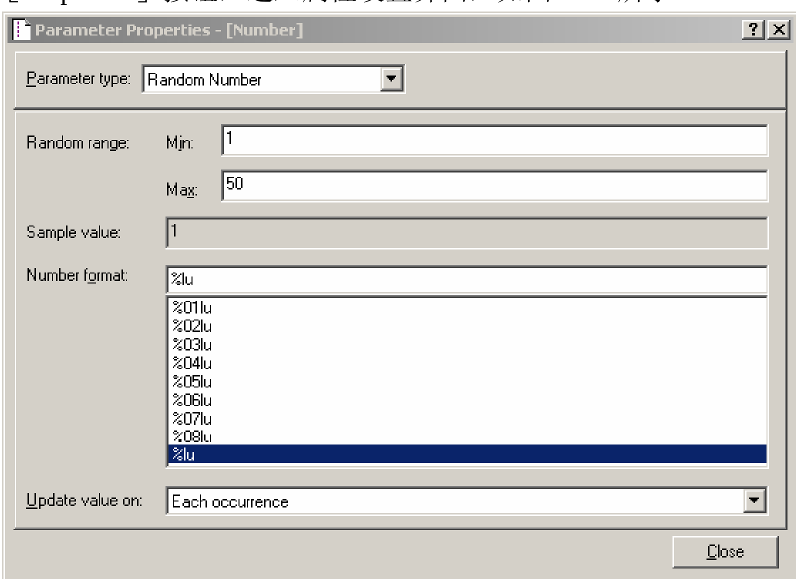


图 3-12　属性设置界面

d．添入随机数的取值范围为（1～50），选择一种数据格式。在 Update Value on 中有以下几个选项：

- Each Occurrence：在运行时，每遇到一次该参数，便会取一个新的值；
- Each iteration：运行时，在每一次循环中都取相同的值；
- Once：运行时，在每次循环中，该参数只取一次值；

在本测试案例中，我们选择“Once”即可，到此参数设置完成，下一步便可执行以上脚本。

③ 测试脚本的执行

a．在 Visual User Generator 工具中，单击菜单 File -> Open 或按“Ctrl＋O”快捷键，打开要调试的测试脚本，如图 3-13 所示。

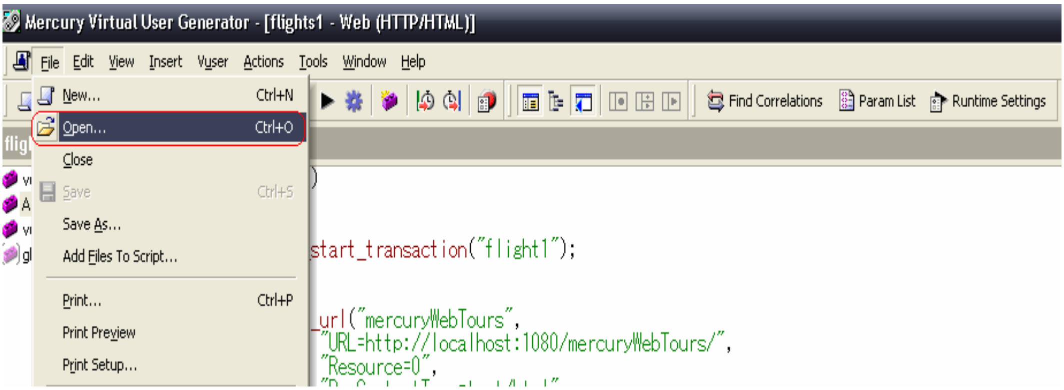


图 3-13　打开要调试的测试脚本菜单

b．单击菜单 Vuser -> Run 或按 F5 快捷键，Visual User Generator 便会执行当前脚本，如图 3-14 所示，如果脚本因语法等错误而无法执行，错误信息将在“Execution Log”栏中显示。

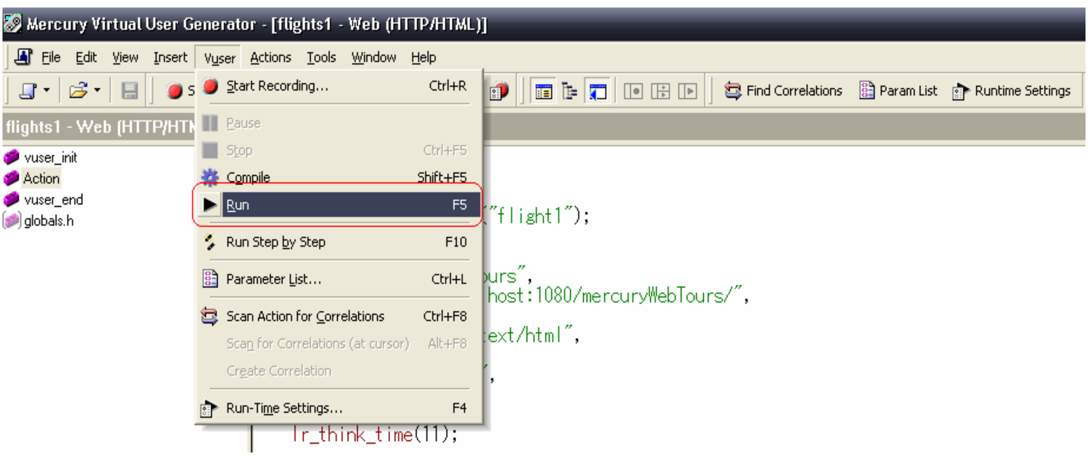


图 3-14　执行测试脚本菜单

c．通过上述步骤将用户名称参数化后，脚本将成功执行，执行结束后自动返回执行结果，如图 3-15 所示。

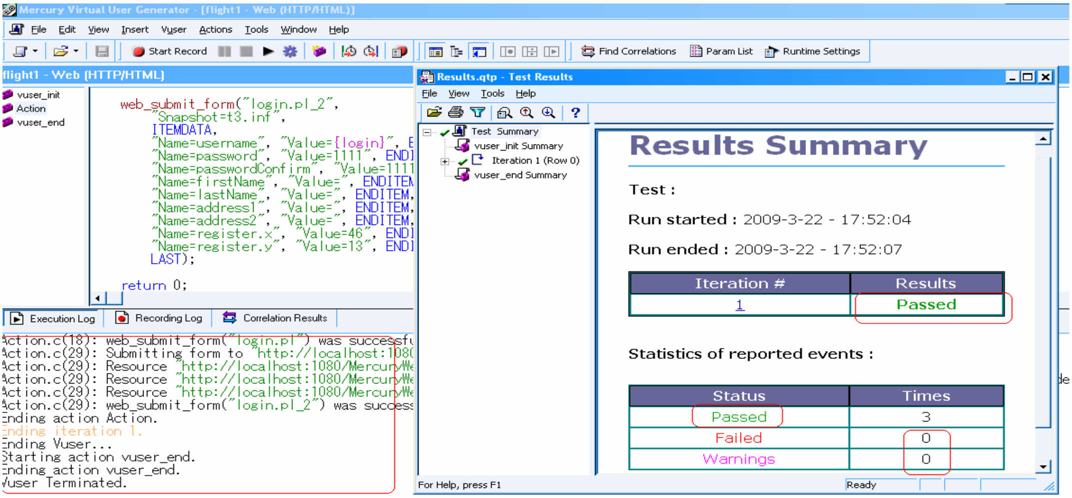


图 3-15　执行结果窗口

以上是测试用例 UserCase_01 的测试执行全部过程。

（2）性能测试用例执行过程

本节将继续以网上订票系统为例，讲解利用 LoadRunner 测试工具进行性能测试的执行过程，具体如下：

① 性能测试目标（见表 3-3）

表 3-3　性能测试目标及用户需求

我们的测试目标	用 户 需 求
事务平均响应时间小于 4s	平均花费 4s 即可完成一次任务
平均每秒完成的事务数量大于 2	每秒能够完成 2 次以上登录任务
测试最优硬件配置、评估新产品	什么样的配置提供了最好的性能
最大虚拟用户并发数为 100	在没有较大性能衰减的前提下，系统能够承受 50 个并发用户负载
在 100 并发虚拟用户下，事务失败率小于 1%	在 100 并发虚拟用户下，事务失败率小于 1%

② 测试环境

a．硬件环境（见表 3-4）

表 3-4　硬件环境

NO.	硬件名称	硬件配置	生产厂商	设备用途
1	普通 PC	CPU：P4 2.0G 内存：1024 MB 硬盘：60 GB	联想	测试终端 安装 LoadRunner
2	PC 服务器	CPU：P4 2.8G×2 内存：1024 MB 硬盘：160 GB	联想	应用服务器 安装被测试系统
3	负载均衡设备	千兆	F5	负载均衡

b．软件环境（表 3-5）

表 3-5　软件环境

NO.	软件名称	版本号	生产厂商	备　　注
1	Win2003Server	2003	MS（微软）	操作系统（应用服务器）
2	WinXP（SP2）	XP	MS（微软）	操作系统（负载发生终端）
3	LoadRunner	8.0	Mercury	测试工具
4	网上订票系统	1.0	Mercury	被测试软件

c．拓扑图（如图 3-16 所示）

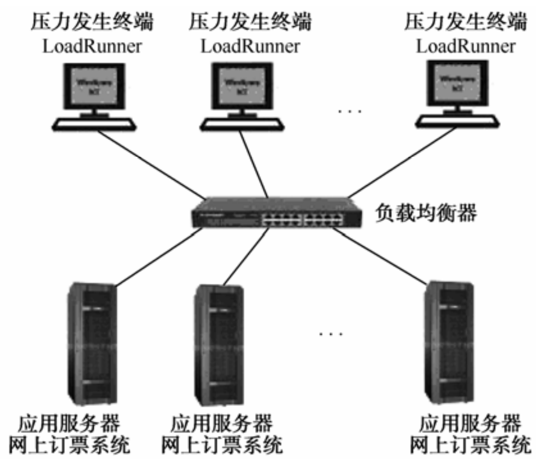


图 3-16　性能测试拓扑图

③ 性能测试脚本

性能测试脚本录制过程，具体参见功能测试用例 UserCase_01 测试脚本的录制。

④ 创建性能测试场景

运行场景描述的是测试活动中发生的各种事件。一个运行场景包括一个运行虚拟用户活动的 Load Generator ——测试终端机器列表、一个测试脚本的列表以及大量的虚拟用户和虚拟用户组。创建运行场景使用 Controller 程序。

a．启动 Controller 程序，将打开“New Scenario”窗口，如图 3-17 所示，如果没有打开此窗口，可以在主菜单或者工具栏中单击“New”。在新建场景的窗口，选择一种场景类型。下面对三种类型进行简单的说明：

- Manual Scenario：该项要完全手动的设置场景。
- Manual Scenario with Percentage Mode：该项只有在“Manual Scenario”选中的情况下才能选择。选择该项后，在场景中我们需要定义要使用的虚拟用户的总数 Load Generator machine 机器集，然后我们为每一个脚本分配要运行的虚拟用户的百分比。
- Goal—Oriented Scenario： 在测试计划中，一般都包括性能测试要达到的目标。选择该项后，LoadRunner 基于这个目标，自动为你创建一个场景。在场景中，我们只要定义好我们的目标即可。

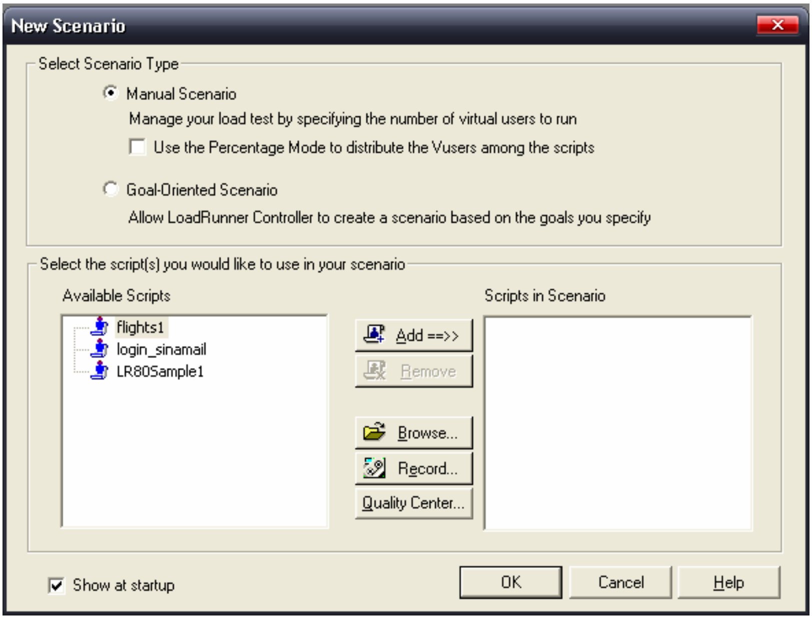


图 3-17　新建场景的窗口

b．在“New Scenario”窗口，还需要为测试场景选择测试脚本，在左侧“Available Scripts”视图中，选中当前测试场景需要的脚本，单击[Add]按钮，脚本便被添加到“Scripts in Scenario”视图中，单击［OK］按钮，创建场景操作完成，将进入场景设置窗口，接下来便可以设置场景了。

⑤ 设置性能测试场景

a．添加压力发生器 Load Generators

在“Scenario Schadule”窗口，单击“Scenario Group”视图右侧的［Generators］按钮，将弹出“Load Generators”子窗口，如图 3-18 所示。

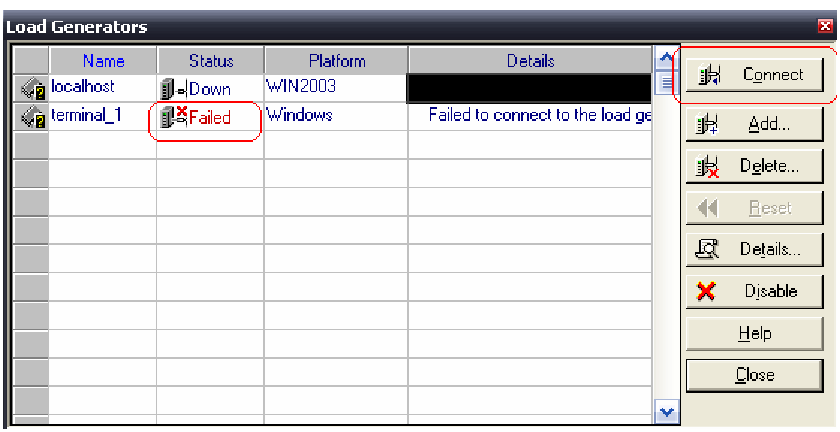


图 3-18　压力生成器窗口

在此子窗口中单击［Add］按钮，将弹出“Add New Load Generator”对话框，如图 3-19 所示。

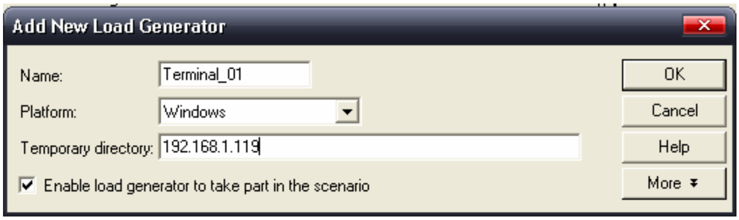


图 3-19　添加压力生成器信息对话框

在此对话框，输入所添加的测试发生器的名称、操作系统平台以及 IP 地址等信息，单击［OK］按钮，测试发生器添加完成，将返回到“Load Generators”子窗口，可以选择添加完的测试发生器，单击［Connect］按钮来测试发生器是否连接畅通，如果 Status 为“Reday”则连接畅通，为“Failed”则表示连接不成功。

注意：这里所添加的测试发生器，机器上必须安装了 LoadRunner Agent Service。

b．添加场景组/设置虚拟用户数量：

在“Scenario Schadule”窗口，如图单击“Scenario Group”视图右侧的［Add Group］按钮，将弹出“Add Group ”对话框，如图 3-20 所示。

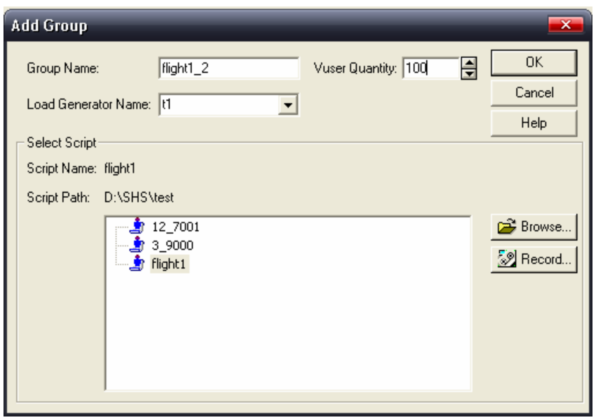


图 3-20　添加场景组对话框

在“Add Group”对话框中，选择一个测试脚本，输入场景组名称，测试发生器的名称及虚拟用户数量等信息，单击［OK］按钮，将在“Scenario Schadule”窗口，“Scenario Group”表格视图中，增加一条场景组记录，如图 3-21 所示，在“Scenario Group”表格视图中，可以修改场景组对应的“Quantity”的值来修改虚拟用户数量。



图 3-21　设置虚拟用户数量窗口

c．设置 Schadule

Schadule 设置是非常重要的，也是三种场景类型最重要的区别之处。

（a）单击［Edit Schadule］按钮，即可进入“Schedule Builder”界面，如图 3-22 所示。

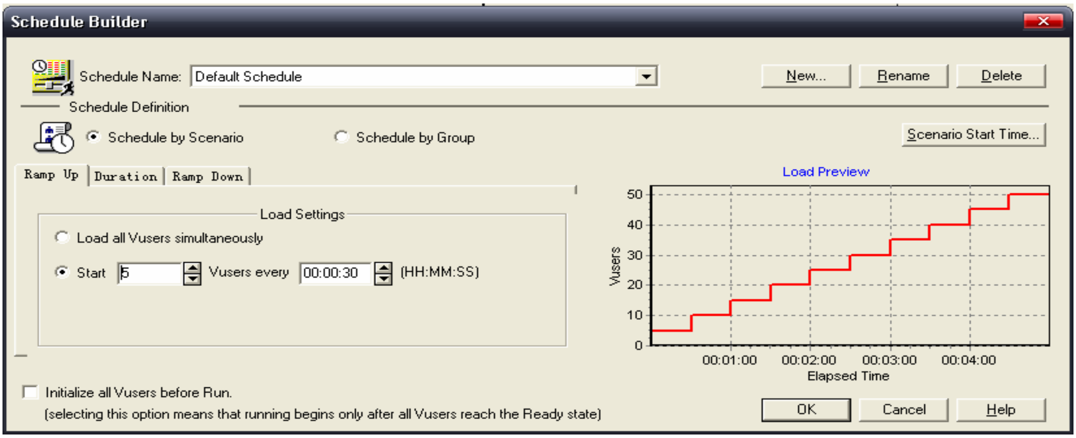


图 3-22　按方案计划加压设置

在“Schedule Builder”窗口中的“Ramp Up”TAB 页，可以设置场景运行时迭代方式启动虚拟用户的参数，具体如下：

- Load all Vusers simultaneously：该选项表示同时加载所有虚拟用户
- Start/Vusers every：该选项表示每 30s 增加 5 个虚拟用户，直至所有虚拟用户全部被加载。

（b）在“Duration”TAB 页，如图 3-23 所示，可设置如下项目：

- Run until completion：此项表示所有虚拟用户只运行一次，然后场景停止。
- Run for：此项表示加载完成后，场景运行的时间。
- Run indefinitely：选择此项表示一直运行场景，直到人为停止运行。
- Initialize all Vusers before Run：选择此复选框则会在运行场景前，初始化所有虚拟用户。

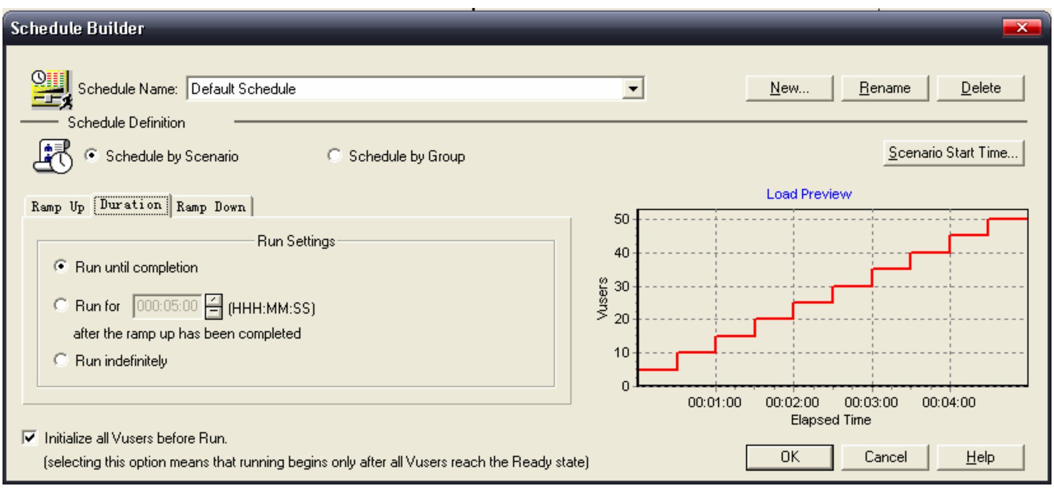


图 3-23　按方案计划的持续时间设置

（c）在“Ramp Down”TAB 页，如图 3-24 所示，可以设置场景运行时迭代方式结束虚拟

用户的参数，具体如下：

- Stop all Vusers simultaneously：该选项表示同时加载所有虚拟用户。
- Stop/Vusers every：该选项表示场景运行结束时，每 30s 结束 2 个虚拟用户，直至所有虚拟用户全部结束。

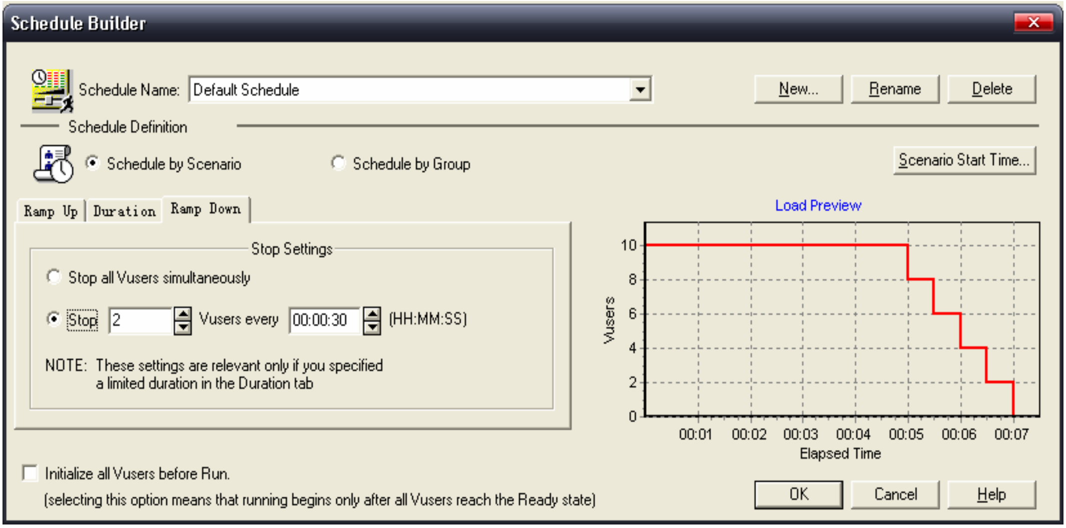


图 3-24　按方案计划减压设置

（d）场景开始运行时间的设置

在“Schedule Builder”窗口中，单击 Scenario Start Time 按钮，将弹出“Scenario Start”对话框，如图 3-25 所示。

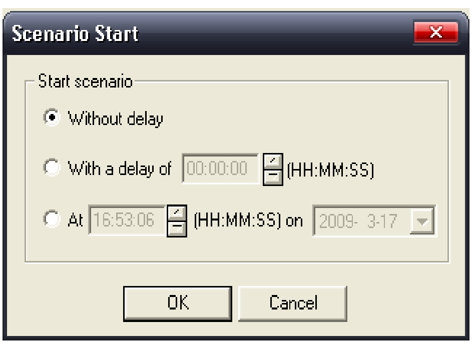


图 3-25　按组计划的延迟时间

在“Scenario Start”对话框中，可设置如下选项：

- Without delay：选择此项，启动场景后，程序会立即运行。
- With a delay of：选择此项，启动场景后，程序会延迟指定的时间后开始运行。
- At［HH:MM:SS］：选择此项，启动场景后，程序会在指定的时刻开始运行。

到此，场景设置完成，接下来就可以在设置好的测试场景中执行性能测试了。

⑥ 执行性能测试

a．在“LoadRunner Controller”主窗口的“RUN” TAB 页面，单击［Start Scenario］按钮，将开始执行设置好的测试场景，如图 3-26 所示。

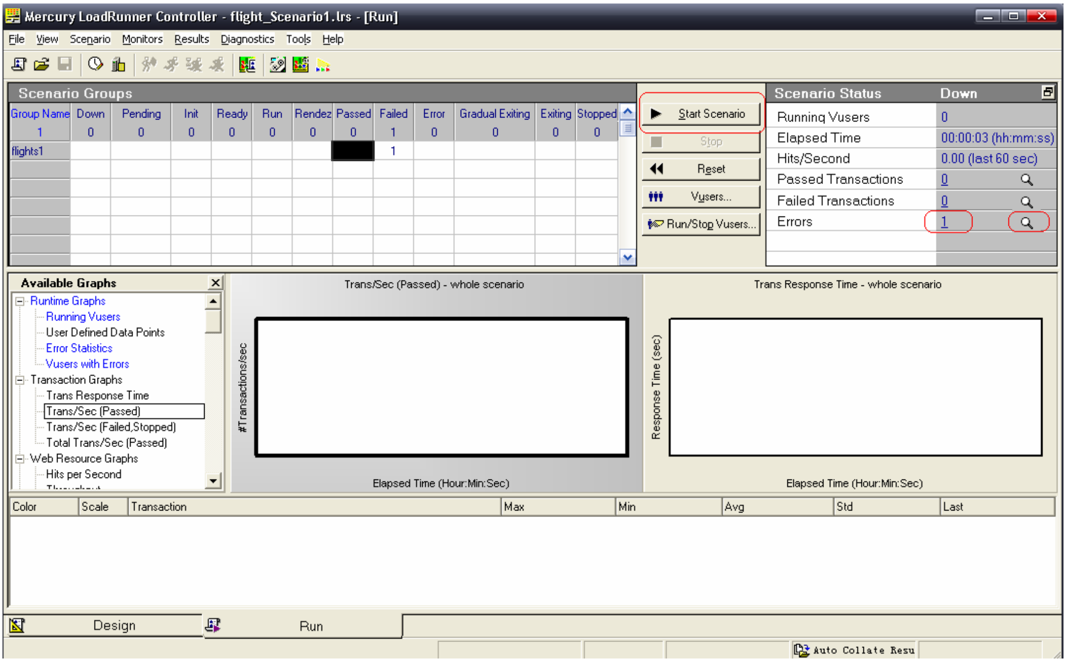


图 3-26　测试场景执行窗口

b. 运行结束后，无论成功还是失败，程序都将返回运行结果，如果运行成功，在“Scenario Groups”视图中的“Passed”字段将显示成功的次数，如图 3-27 所示，否则在“Failed”字段将显示失败的次数，且在“Scenario Status”视图的“Errors”项中将显示返回的错误数量，如图 3-28 所示。

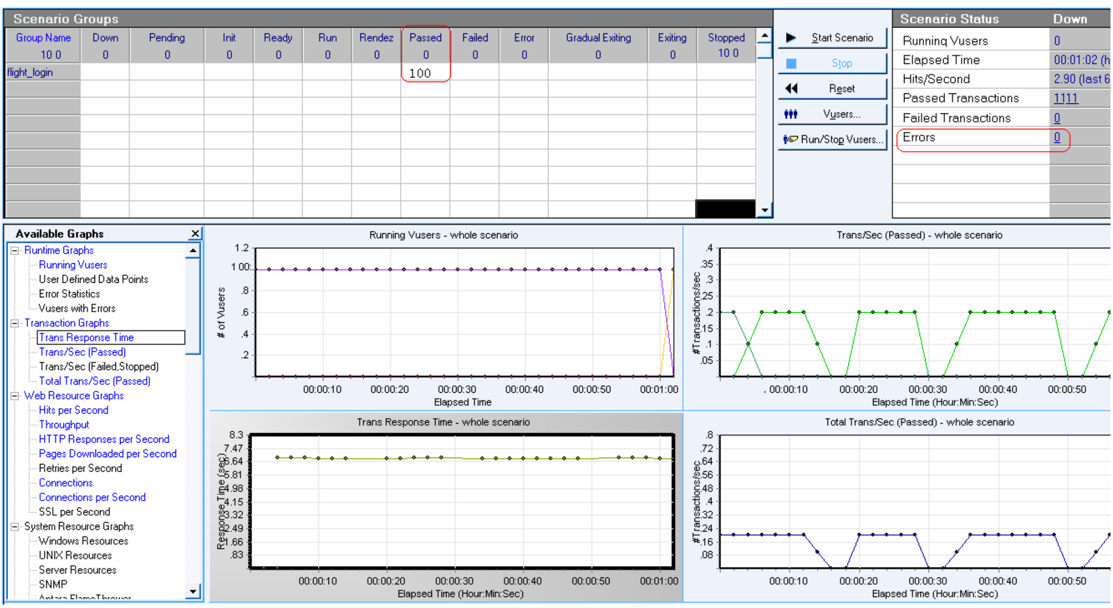


图 3-27　运行成功窗口

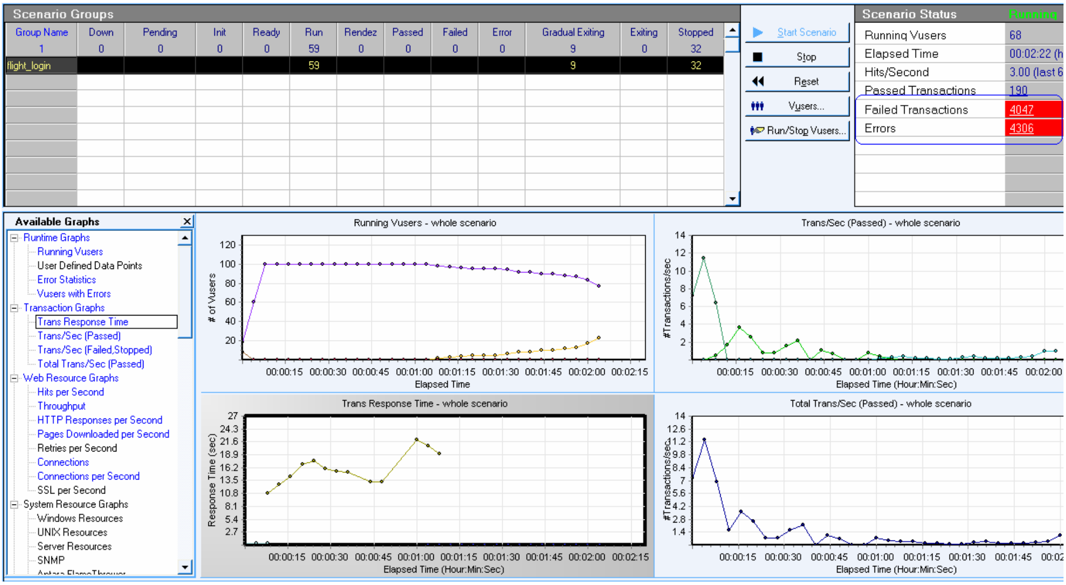


图 3-28　运行失败窗口

依照上述步骤执行自动化测试设计中的所有测试场景，并记录测试结果，再将测试结果写入测试报告，自动化测试的执行到此就全部完成。

3.2.4　应知应会

1．软件回归测试

以下文字描述了软件回归测试的概念和进行回归测试的基本步骤，介绍了可用于回归测试的测试用例库的维护方法，给出了几种可以保证回归测试效率和有效性的回归测试策略，总结了回归测试时应该注意的一些实际问题。

（1）概述

在软件生命周期中的任何一个阶段，只要软件发生了改变，就可能给该软件带来问题。软件的改变可能是源于发现了错误并做了修改，也有可能是因为在集成或维护阶段加入了新的模块。当软件中所含错误被发现时，如果错误跟踪与管理系统不够完善，就可能会遗漏对这些错误的修改；而开发者对错误理解得不够透彻，也可能导致所做的修改只修正了错误的外在表现，而没有修复错误本身，从而造成修改失败；修改还有可能产生副作用从而导致软件未被修改的部分产生新的问题，使本来工作正常的功能产生错误。同样，在有新代码加入软件的时候，除了新加入的代码中有可能含有错误外，新代码还有可能对原有的代码带来影响。因此，每当软件发生变化时，我们就必须重新测试现有的功能，以便确定修改是否达到了预期的目的，检查修改是否损害了原有的正常功能。同时，还需要补充新的测试用例来测试新的或被修改了的功能。为了验证修改的正确性及其影响就需要进行回归测试。

回归测试在软件生命周期中扮演着重要的角色，因忽视回归测试而造成严重后果的例子不计其数，导致阿里亚娜 5 型火箭发射失败的软件缺陷就是由于复用的代码没有经过充分的回归测试造成的。

回归测试作为软件生命周期的一个组成部分，在整个软件测试过程中占有很大的工作量比重，软件开发的各个阶段都会进行多次回归测试。在渐进和快速迭代开发中，新版本的连

续发布使回归测试进行得更加频繁，而在极端编程方法中，更是要求每天都进行若干次回归测试。因此，通过选择正确的回归测试策略来改进回归测试的效率和有效性是非常有意义的。

（2）回归测试策略

对于一个软件开发项目来说，项目的测试组在实施测试的过程中会将所开发的测试用例保存到“测试用例库”中，并对其进行维护和管理。当得到一个软件的基线版本时，用于基线版本测试的所有测试用例就形成了基线测试用例库。在需要进行回归测试的时候，就可以根据所选择的回归测试策略，从基线测试用例库中提取合适的测试用例组成回归测试包，通过运行回归测试包来实现回归测试。保存在基线测试用例库中的测试用例可能是自动测试脚本，也有可能是测试用例的手工实现过程。

回归测试需要时间、经费和人力来计划、实施和管理。为了在给定的预算和进度下，尽可能有效地进行回归测试，需要对测试用例库进行维护并依据一定的策略选择相应的回归测试包。

① 测试用例库的维护

为了最大限度地满足客户的需要和适应应用的要求，软件在其生命周期中会频繁地被修改和不断推出新的版本，修改后的或者新版本的软件会添加一些新的功能或者在软件功能上产生某些变化。随着软件的改变，软件的功能和应用接口以及软件的实现发生了演变，测试用例库中的一些测试用例可能会失去针对性和有效性，而另一些测试用例可能会变得过时，还有一些测试用例将完全不能运行。为了保证测试用例库中测试用例的有效性，必须对测试用例库进行维护。同时，被修改的或新增添的软件功能，仅仅靠重新运行以前的测试用例并不足以揭示其中的问题，有必要追加新的测试用例来测试这些新的功能或特征。因此，测试用例库的维护工作还应包括开发新测试用例，这些新的测试用例用来测试软件的新特征或者覆盖现有测试用例无法覆盖的软件功能或特征。

测试用例的维护是一个不间断的过程，通常可以将软件开发的基线作为基准，维护的主要内容包括下述几个方面。

a．删除过时的测试用例

因为需求的改变等原因可能会使一个基线测试用例不再适合被测试系统，这些测试用例就会过时。例如，某个变量的界限发生了改变，原来针对边界值的测试就无法完成对新边界测试。所以，在软件的每次修改后都应进行相应的过时测试用例的删除。

b．改进不受控制的测试用例

随着软件项目的进展，测试用例库中的用例会不断增加，其中会出现一些对输入或运行状态十分敏感的测试用例。这些测试不容易重复且结果难以控制，会影响回归测试的效率，需要进行改进，使其达到可重复和可控制的要求。

c．删除冗余的测试用例

如果存在两个或者更多个测试用例针对一组相同的输入和输出进行测试，那么这些测试用例是冗余的。冗余测试用例的存在降低了回归测试的效率。所以需要定期地整理测试用例库，并将冗余的用例删除掉。

d．增添新的测试用例

如果某个程序段、构件或关键的接口在现有的测试中没有被测试，那么应该开发新测试用例重新对其进行测试，并将新开发的测试用例合并到基线测试包中。

通过对测试用例库的维护不仅改善了测试用例的可用性，而且也提高了测试库的可信性，

同时还可以将一个基线测试用例库的效率和效用保持在一个较高的级别上。

② 回归测试包的选择

在软件生命周期中，即使一个得到良好维护的测试用例库也可能变得相当大，这使每次回归测试都重新运行完整的测试包变得不切实际。一个完全的回归测试包括每个基线测试用例，时间和成本约束可能阻碍运行这样一个测试，有时测试组不得不选择一个缩减的回归测试包来完成回归测试。

回归测试的价值在于它是一个能够检测到回归错误的受控实验。当测试组选择缩减的回归测试时，有可能删除了将揭示回归错误的测试用例，消除了发现回归错误的机会。然而，如果采用了代码相依性分析等安全的缩减技术，就可以决定哪些测试用例可以被删除而不会让回归测试的意图遭到破坏。

选择回归测试策略应该兼顾效率和有效性两个方面。常用的选择回归测试的方式包括：

a．再测试全部用例

选择基线测试用例库中的全部测试用例组成回归测试包，这是一种比较安全的方法，再测试全部用例具有最低的遗漏回归错误的风险，但测试成本最高。全部再测试几乎可以应用到任何情况下，基本上不需要进行分析和重新开发，但是，随着开发工作的进展，测试用例不断增多，重复原先所有的测试将带来很大的工作量，往往超出了我们的预算和进度。

b．基于风险选择测试

可以基于一定的风险标准来从基线测试用例库中选择回归测试包。首先运行最重要的、关键的和可疑的测试，而跳过那些非关键的、优先级别低的或者高稳定的测试用例，这些用例即便可能测试到缺陷，这些缺陷的严重性也仅有三级或四级。一般而言，测试从主要特征到次要特征。

c．基于操作剖面选择测试

如果基线测试用例库的测试用例是基于软件操作剖面开发的，测试用例的分布情况反映了系统的实际使用情况。回归测试所使用的测试用例个数可以由测试预算确定，回归测试可以优先选择那些针对最重要或最频繁使用功能的测试用例，释放和缓解最高级别的风险，有助于尽早发现那些对可靠性有最大影响的故障。这种方法可以在一个给定的预算下最有效地提高系统可靠性，但实施起来有一定的难度。

d．再测试修改的部分

当测试者对修改的局部化有足够的信心时，可以通过相依性分析识别软件的修改情况并分析修改的影响，将回归测试局限于被改变的模块和它的接口上。通常，一个回归错误一定涉及一个新的、修改的或删除的代码段。在允许的条件下，回归测试尽可能覆盖受到影响的部分。

再测试全部用例的策略是最安全的策略，但已经运行过许多次的回归测试不太可能揭示新的错误，而且很多时候，由于时间、人员、设备和经费的原因，不允许选择再测试全部用例的回归测试策略，此时，可以选择适当的策略进行缩减的回归测试。

③ 回归测试的基本过程

有了测试用例库的维护方法和回归测试包的选择策略，回归测试可遵循下述基本过程进行：

a．识别出软件中被修改的部分；

b．从原基线测试用例库 T 中，排除所有不再适用的测试用例，确定那些对新的软件版本

依然有效的测试用例，其结果是建立一个新的基线测试用例库 T0。

c．依据一定的策略从 T0 中选择测试用例测试被修改的软件。

d．如果必要，生成新的测试用例集 T1，用于测试 T0 无法充分测试的软件部分。

e．用 T1 执行修改后的软件。

第（2）和第（3）步测试验证修改是否破坏了现有的功能，第（4）和第（5）步测试验证修改工作本身。

（3）回归测试实践

在实际工作中，回归测试需要反复进行，当测试者一次又一次地完成相同的测试时，这些回归测试将变得非常令人厌烦，而在大多数回归测试需要手工完成的时候尤其如此，因此，需要通过自动测试来实现重复的和一致的回归测试。通过测试自动化可以提高回归测试效率。为了支持多种回归测试策略，自动测试工具应该是通用的和灵活的，以便满足达到不同回归测试目标的要求。

在测试软件时，应用多种测试技术是常见的。当测试一个修改了的软件时，测试者也可能希望采用多于一种回归测试策略来增加对修改软件的信心。不同的测试者可能会依据自己的经验和判断选择不同的回归测试技术和策略。

回归测试并不减少对系统新功能和特征的测试需求，回归测试包应包括新功能和特征的测试。如果回归测试包不能达到所需的覆盖要求，必须补充新的测试用例使覆盖率达到规定的要求。

回归测试是重复性较多的活动，容易使测试者感到疲劳和厌倦，降低测试效率，在实际工作中可以采用一些策略减轻这些问题。例如，安排新的测试者完成手工回归测试，分配更有经验的测试者开发新的测试用例，编写和调试自动测试脚本，做一些探索性的或 ad hoc 测试。还可以在不影响测试目标的情况下，鼓励测试者创造性地执行测试用例，变化的输入、按键和配置能够有助于激励测试者又能揭示新的错误。

在组织回归测试时需要注意两点，首先是各测试阶段发生的修改一定要在本测试阶段内完成回归，以免将错误遗留到下一测试阶段。其次，回归测试期间应对该软件版本冻结，将回归测试发现的问题集中修改，集中回归。

在实际工作中，可以将回归测试与兼容性测试结合起来进行。在新的配置条件下运行旧的测试可以发现兼容性问题，而同时也可以揭示编码在回归方面的错误。

2．性能测试

（1）为什么进行性能测试

目的是验证软件系统是否能够达到用户提出的性能指标，同时发现软件系统中存在的性能瓶颈，优化软件，最后起到优化系统的目的。

它包括以下几个方面：

① 评估系统的能力，测试中得到的负荷和响应时间数据可以被用于验证所计划的模型的能力，并帮助做出决策。

② 识别体系中的弱点：受控的负荷可以被增加到一个极端的水平，并突破它，从而修复体系的瓶颈或薄弱的地方。

③ 系统调优：重复运行测试，验证调整系统的活动得到了预期的结果，从而改进性能。

检测软件中的问题：长时间的测试执行可导致程序发生由于内存泄露引起的失败，揭示程序中隐含的问题或冲突。

④ 验证稳定性（resilience）可靠性（reliability）：在一个生产负荷下执行测试一定的时间是评估系统稳定性和可靠性是否满足要求的唯一方法。

（2）性能测试主要类型

① 负载测试：负载测试是一种性能测试，指数据在超负荷环境中运行，程序是否能够承担。

② 强度测试：强度测试是一种性能测试，它测试在系统资源特别低的情况下软件系统运行的情况。

③ 容量测试：确定系统可处理同时在线的最大用户数。

3.2.5　拓展任务

独立执行天天超市管理系统的测试执行过程。

工作任务 3.3　测试执行结果与分析

3.3.1　网上订票系统的测试结果与分析

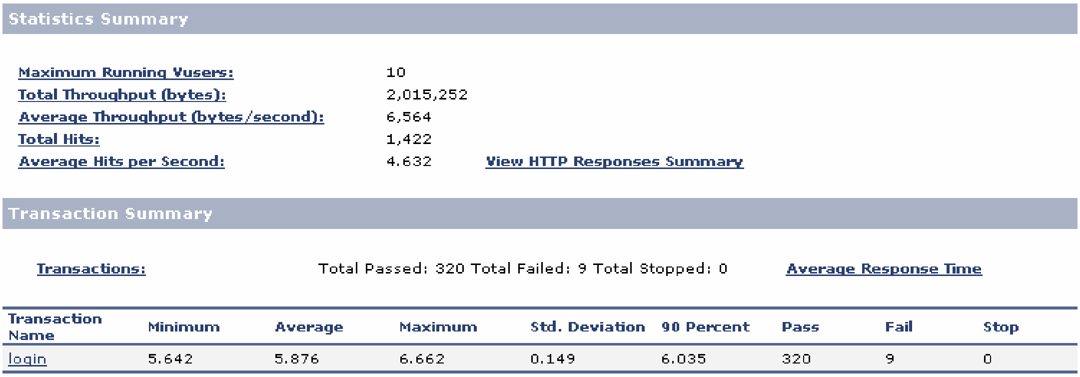
1．工作任务描述

以网上订票系统的一次性能测试为例，10 个并发虚拟用户，测试执行时间为 5min，具体测试结果如下：

2．工作过程

（1）数据统计概要

如图 3-29 所示，数据概要结果显示，事务响应时间最小值为 5.642s，最大值为 6.662s，平均值为 5.876s。


Statistics Summary

Maximum Running Vusers:	10	
Total Throughput (bytes):	2,015,252	
Average Throughput (bytes/second):	6,564	
Total Hits:	1,422	
Average Hits per Second:	4.632	View HTTP Responses Summary

Transaction Summary

Transactions: Total Passed: 320 Total Failed: 9 Total Stopped: 0 Average Response Time

Transaction Name	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
login	5.642	5.876	6.662	0.149	6.035	320	9	0

图 3-29　数据概要结果窗口

（2）完成事务总数

如图 3-30 所示，本次测试中成功事务的总数为 320，错误事务总数为 9。

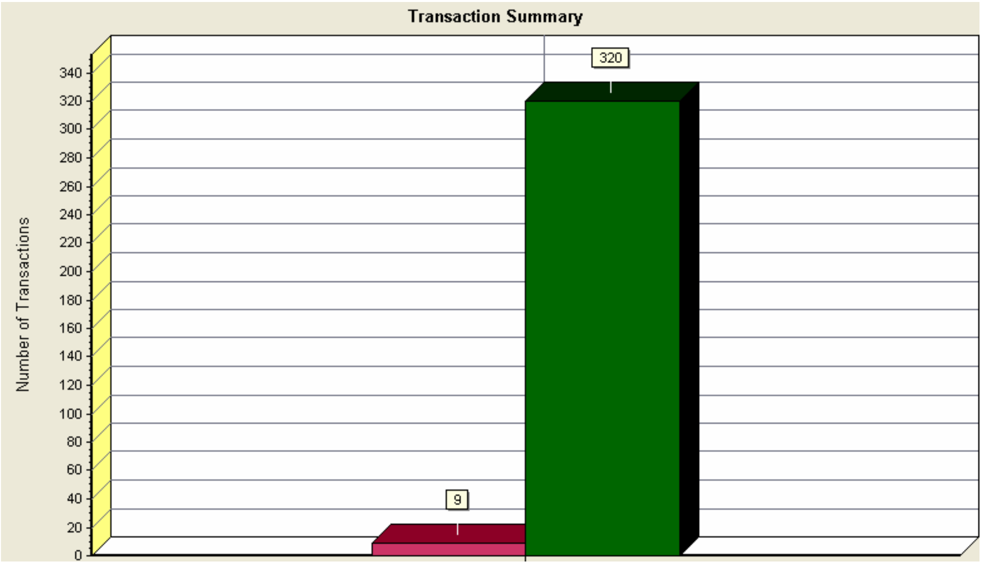


图 3-30　成功事务总数窗口

（3）平均事务响应时间（秒）

如图 3-31 所示，所显示的结果是事务的响应时间，由图表中曲线数据可以看出事务响应时间分布在 5.75～6.17s 之间，比较平稳。

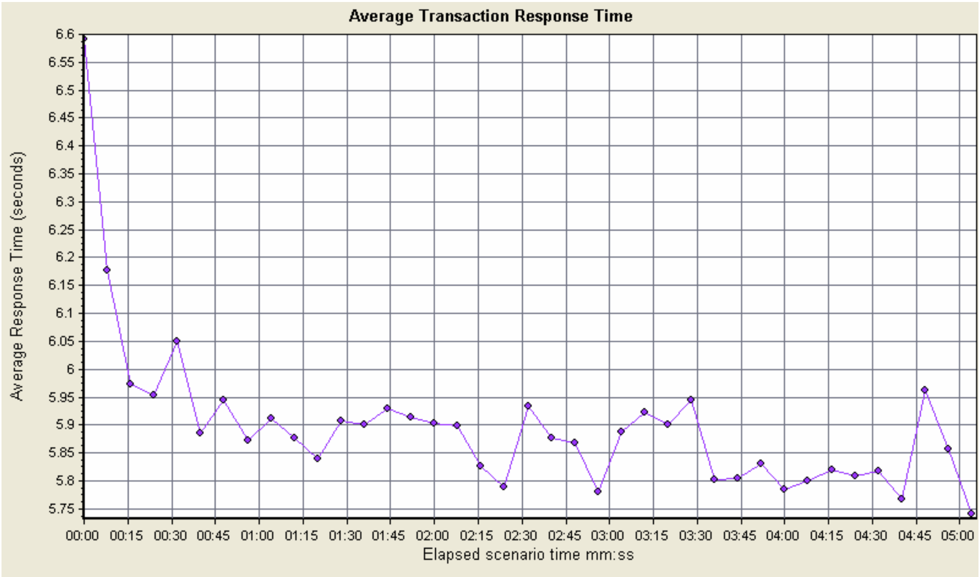


图 3-31　事务的响应时间窗口

（4）平均每秒完成事务数

表格中上方曲线为表示平均每秒成功的事务，下方曲线表示错误的事务，如图 3-32 所示。

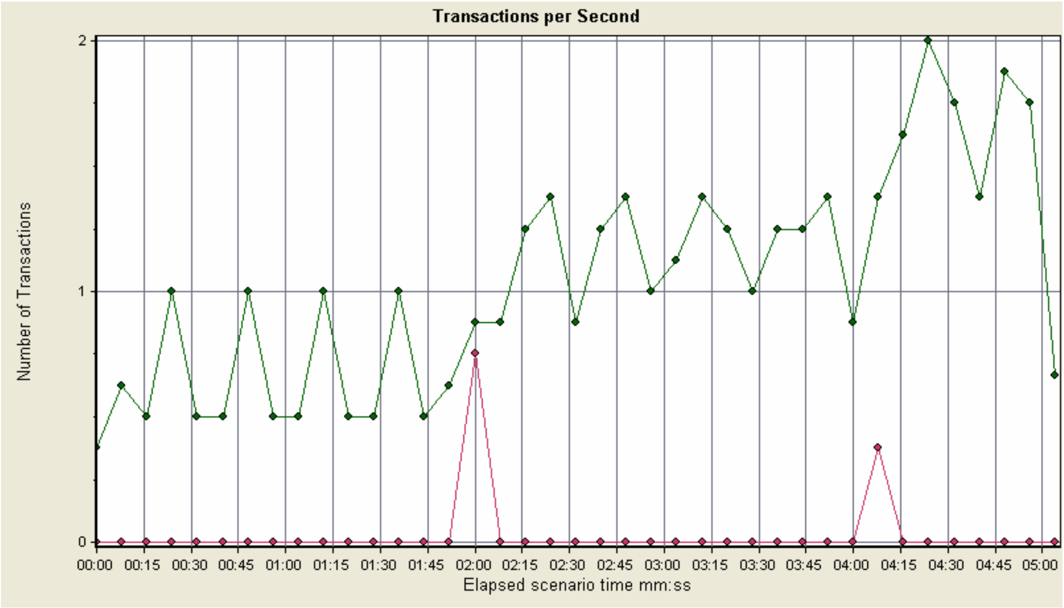


图 3-32　平均每秒完成事务数窗口

（5）负载发生终端资源使用情况，如图 3-33 所示。

1	CPU使用率	可用内存(Kbytes)	网络吞吐量(Bytes/sec)	硬盘I/O使用情况
2	23.308	60.2	15.749	13.33
3	12.587	61.389	1.213	0
4	19.011	61.287	1.039	8.91
5	16.276	61.25	0.995	0
6	11.979	61.289	0.939	0
7	19.531	61.273	1.055	17.817
8	14.844	61.156	0.997	3.325
9	10.59	61.232	0.874	2.223
10	18.576	61.272	1.298	0
11	11.458	61.266	0.903	0

图 3-33　负载发生终端资源使用情况

（6）应用服务器资源使用情况如图 3-34 所示。

1	CPU使用率	可用内存(Kbytes)	网络吞吐量(Bytes/sec)	硬盘I/O使用情况
2	73.308	40.2	105.749	23.33
3	82.587	41.379	117.179	17.53
4	90.041	41.266	125.016	8.91
5	86.276	41.25	131.433	8.94
6	91.929	41.189	135.235	11.54
7	89.631	41.253	127.711	17.814
8	94.044	41.256	118.712	3.325
9	90.59	41.232	124.661	5.223
10	88.578	41.277	124.525	9.263
11	91.258	41.266	121.248	20.263

图 3-34　应用服务器资源使用情况

3．测试结果分析

由以上测试数据可以看出：

1．用 10 个并发虚拟用户进行测试时，服务器返回了错误结果，错误发生率为 2.7%。

2．负载发生终端机器资源使用很小，应用服务器 CPU 资源占用较大，说明在 10 个并发用户状态下，应用服务器处理事务能力达到极限，被诊断为性能的瓶颈所在。

4．测试结论

由于应用服务器的 CPU 处理能力较差，而不能满足用户的性能需要。

5．性能优化建议

提高应用服务器的处理能力。

3.3.2 应知应会

软件测试执行结束后，测试活动还没有结束。测试结果分析是必不可少的重要环节，“编筐编篓，全在收口”，测试结果的分析对下一轮测试工作的开展有很大的借鉴意义。测试结束后，也应该分析自己发现的软件缺陷，对发现的缺陷分类，你会发现自己提交的问题只有固定的几个类别；然后，再把一起完成测试执行工作的其他测试人员发现的问题也汇总起来，你会发现，你所提交问题的类别与他们有差异。这很正常，人的思维是有局限性的，在测试的过程中，每个测试人员都有自己思考问题的盲区和测试执行的盲区，有效地自我分析和分析其他测试人员，你会发现自己的盲区，有针对性地分析盲区，必定会在下一轮测试中避免盲区。

3.3.3 拓展任务

独立分析天天超市管理系统执行的结果。

第 4 章　测 试 总 结

本章介绍网上购物系统的测试总结的编写。

本章重点：

- 测试总结编写规范
- 测试总结与测试报告的区别

测试总结是在每做完一轮测试之后要写的一个总结性质的文档。写测试总结是很有必要的，因为每做完一轮测试如果能够详细地总结这次测试过程中发现的问题及要注意的事项，就能在下一轮测试中更有效地选择测试的方向。

工作任务 4.1 测试总结

4.1.1　网上购物系统的测试总结

1．工作任务描述

网上购物系统是 Web 应用程序，而 Web 测试相对于非 Web 测试来说都是更具挑战性的工作，因为用户对 Web 页面质量有很高的期望。所以在每个版本测试（回归测试）完成后，都要进行测试总结，包括列出 Bug 的严重级别及比例，总结人员的工作效率，对风险进行评估，并且给出下一版本的建议等。

在前面编写测试计划、设计测试用例并执行测试用例的基础上，本章任务就是撰写网上购物系统的测试总结。

2．工作过程

（1）网上购物系统的测试总结

下面是网上购物系统的测试总结报告。

网上购物系统测试总结报告

<table>
<tr><td>文档标识：</td><td colspan="2"></td><td colspan="2">当前版本：</td><td></td></tr>
<tr><td rowspan="2">当前状态：</td><td>草稿</td><td></td><td colspan="2" rowspan="2">发布日期：</td><td rowspan="2"></td></tr>
<tr><td>发布</td><td></td></tr>
<tr><td colspan="6">修改历史</td></tr>
<tr><td>日期</td><td>版本</td><td>作者</td><td>修改内容</td><td>评审号</td><td>变更控制号</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

目　录

一、测试概述
1.1　编写目的
1.2　测试范围
1.3　参考资料
二、测试计划执行情况
2.1　测试类型
2.2　进度偏差
2.3　测试环境与配置
2.4　测试机构和人员
2.5　测试问题总结
三、测试总结
3.1　测试用例执行结果
3.2　测试问题解决
3.3　测试结果分析
3.3.1　覆盖分析
3.3.2　缺陷分析
四、综合评价
4.1　软件能力
4.2　建议

一、测试概述

1.1　编写目的

对网上购物系统项目中所有的软件测试活动中，包括测试进度、资源、问题、风险以及测试组和其他组间的协调等进行评估，总结测试活动的成功经验与不足，以便今后更好地开展测试工作。

本系统测试总结报告的预期读者是：

开发部经理；
项目组所有人员；
测试组人员；
SQA 人员；
SCM 人员。

以及长春职业技术学院信息分院软件教研室软件测试组授权调阅本文档的其他人员。

1.2　测试范围

网上购物系统项目因其自身的特殊性，测试组仅依据用户需求说明书和软件需求规格说明书以及相应的设计文档进行系统测试，包括功能测试、性能测试、用户访问与安全控制测试、用户界面测试以及兼容性测试等，而单元测试和集成测试则由开发人员来执行。主要功能包括：

前台用户功能
用户登录
注册基本信息

浏览商品
查询商品
购物车管理
订单管理
后台信息管理功能
管理员登录
商品管理模块
订单管理模块
用户管理模块

1.3 参考资料

资料名称	版　本	作　者	是否经过评审	备　注
网上购物系统软件开发计划.doc	2.0		已评审	
网上购物系统测试方案.doc	1.1		已评审	
网上购物测试计划.doc	1.1		已评审	
网上购物测试进度表.mpp	1.1		已评审	

二、测试计划执行情况

2.1 测试类型

测试类型	测 试 内 容	测 试 目 的	所用的测试工具和方法
功能测试	用户前台：用户登录、注册基本信息、浏览商品、查询商品、购物车管理、订单管理 管理后台：管理员登录系统、商品管理模块、订单管理模块、用户管理模块	核实所有功能均已正常实现，即可按每个用户的需求购买商品 1．业务流程检验：各个业务流程符合常规逻辑，用户使用时不会产生疑问 2．数据精确：各数据类型的输入输出时统计精确	采用黑盒测试，使用边界值测试、等价类划分、数据驱动等测试方法，进行手工测试
用户界面（UI）测试	1．导航、链接、Cookie、页面结构包括菜单、背景、颜色、字体、按钮名称、TITLE、提示信息的一致性等 2．友好性、易用性、合理性、一致性、正确性等（详见网站）	核实各个窗口风格（包括颜色、字体、提示信息、图标、TITLE 等等）都与基准版本保持一致，或符合可接受标准，能够保证用户界面的友好性、易操作性，而且符合用户操作习惯	Web 测试通用方法 手工测试
安全性和访问控制测试	1．密码：登录、个人用户、管理员用户 2．权限限制 3．通过修改 URL 非法访问 4．登录超时限制等	1．应用程序级别的安全性：核实用户只能操作其所拥有权限能操作的功能 2．系统级别的安全性：核实只有具备系统访问权限的用户才能访问系统	黑盒测试、手工测试

续表

测试类型	测 试 内 容	测 试 目 的	所用的测试工具和方法
兼容性测试	1．用不同版本的不同浏览器：NetScape、MyIE、Tecent，IE5.5，IE6.0，分辨率：800×600、1024×768，操作系统：WIN 2000 Server、WIN 2000 Professional、WIN XP 分别进行测试 2．不同操作系统、浏览器、分辨率和各种运行软件等各种条件的组合测试	核实系统在不同的软件和硬件配置中运行稳定	黑盒测试、手工测试
性能测试	1．最大并发数 2．查询商品、加入购物车时，注册新用户时以及登录时系统的响应时间	核实系统在大流量的数据与多用户操作时软件性能的稳定性，不造成系统崩溃或相关的异常现象	LoadRunner 8.0 自动化测试

2.2 进度偏差

测试活动	计划起止日期	实际起止日期	进度偏差	备 注
制定测试计划				待 SDP 评审完毕
测试计划评审				等待和 SCMP、SQAP 同时评审
分解测试需求				
测试需求 Review				
选定测试范围				
编写测试方案				
测试方案评审				待测试用例设计完毕后评审
设计测试用例				根据需求变更修改用例
测试用例评审				
测试执行				测试移交延迟一天
测试总结				

2.3 测试环境与配置

资源名称/类型	配 置
测试 PC（4 台）	P4，主频 1.6GHz 以上，硬盘 40GB，内存 512MB，本要求是最小配置
TD7.6 服务器，DB 服务器（同 1 台）	PC Server：512MB 内存、40GB SCSI 硬盘
数据库管理系统	SQL Server 2000
应用软件	Microsoft Office、Visio、Visual Sourcesafe、Microsoft Project
客户端前端展示	IE 6.0
负载性能测试工具	Load Runner 8.0
功能性测试工具	Manual
测试管理工具	TestDirector 7.6

2.4　测试机构和人员

测试阶段	测试机构名称	负责人	参与人员	所充当角色
系统测试	软件教研室测试组	于艳华	于艳华、王素华	测试人员

2.5　测试问题总结

在整个系统测试执行期间，项目组开发人员高效地及时解决测试组人员提出的各种缺陷，在一定程度上较好地保证了测试执行的效率以及测试最终期限。但是在整个软件测试活动中还是暴露了一些问题，表现在：

测试执行时间相对较少，测试通过标准要求较低；

测试执行人员对管理平台不够熟悉，使用时效率偏低；

测试执行人员对系统了解不透彻，测试执行时存在理解偏差，导致提交无效缺陷。

三、测试总结

3.1　测试用例执行结果

测试用例 标识号	测试用例 名称	用 例 状 态	测 试 结 果	备　　注
用户管理部分				
Test Case 001	必添项是否允许为空	已执行	测试通过	添加注册信息
Test Case 002	必添项仅输入空格	已执行	测试通过	
Test Case 003	输入字符数等于域允许的最大字符数	已执行	测试通过	
Test Case 004	输入字符数大于域允许的最大字符数	已执行	测试通过	
Test Case 005	tab 校验	已执行	测试通过	
Test Case 006	用户名中包含空格	已执行	测试通过	
Test Case 007	特殊字符校验	已执行	测试通过	
Test Case 008	密码校验	已执行	测试通过	
Test Case 009	“用户名”重名校验	已执行	测试通过	
测试用例 标识号	测试用例 名称	用例状态	测试结果	备注
Test Case 010	“姓名”重名校验	已执行	测试通过	
Test Case 011	回车验证	已执行	测试通过	
Test Case 012	过期校验	已执行	测试通过	
Test Case 013	密码显示校验	已执行	测试通过	
Test Case 014	用户名大小写校验	已执行	测试通过	
Test Case 015	页面切换校验	已执行	测试通过	
Test Case 016	回车验证	已执行	测试通过	管理员登录
Test Case 017	登录次数校验	已执行	测试通过	
Test Case 018	权限校验	已执行	测试通过	
Test Case 019	注入式登录	已执行	测试通过	
Test Case 020	注册用户登录	已执行	测试通过	

续表

测试用例标识号	测试用例名称	用例状态	测试结果	备注
Test Case 021	锁定用户登录	已执行	测试通过	
Test Case 022	tab 校验	已执行	测试通过	
Test Case 023	回车验证	已执行	测试通过	注册用户登录
Test Case 024	登录密码中包含空格	已执行	测试通过	
Test Case 025	登录密码大小写校验	已执行	测试通过	
Test Case 026	登录次数校验	已执行	测试通过	
Test Case 027	使用字符长度等于临界值的用户名和密码登录	已执行	测试通过	
Test Case 028	使用含有空格的用户名登录	已执行	测试通过	
Test Case 029	使用超长用户名和密码登录	已执行	测试通过	
Test Case 030	注入式登录	已执行	测试通过	
Test Case 031	锁定用户登录	已执行	测试通过	
Test Case 032	tab 校验	已执行	测试通过	
Test Case 033	必添项是否允许为空	已执行	测试通过	修改注册信息
Test Case 034	必添项仅输入空格	已执行	测试通过	
Test Case 035	输入字符数大于域允许的最大字符数	已执行	测试通过	
Test Case 036	输入字符数等于域允许的最大字符数	已执行	测试通过	
Test Case 037	tab 校验	已执行	测试通过	
Test Case 038	用户名中包含空格	已执行	测试通过	
Test Case 039	不修改直接保存	已执行	测试通过	
测试用例标识号	测试用例名称	用例状态	测试结果	备注
Test Case 040	特殊字符校验	已执行	测试通过	
Test Case 041	密码校验	已执行	测试通过	
Test Case 042	“用户名”重名校验	已执行	测试通过	
Test Case 043	回车验证	已执行	测试通过	
Test Case 044	页面切换校验	已执行	测试通过	
Test Case 045	过期校验	已执行	测试通过	
Test Case 046	密码显示校验	已执行	测试通过	
商品管理部分				
Test Case 047	必添项是否允许为空	已执行	测试通过	商品类别管理

续表

测试用例标识号	测试用例名称	用例状态	测试结果	备注
Test Case 048	输入字符数等于域允许的最大字符数	已执行	测试通过	
Test Case 049	输入字符数大于域允许的最大字符数	已执行	测试通过	
Test Case 050	回车验证	已执行	测试通过	
Test Case 051	验证系统定义的域长度是否够用	已执行	测试通过	
Test Case 052	重名校验	已执行	测试通过	
Test Case 053	必添项是否允许为空	已执行	测试通过	
Test Case 054	输入字符数等于域允许的最大字符数	已执行	测试通过	
Test Case 055	输入字符数大于域允许的最大字符数	已执行	测试通过	
Test Case 056	不修改直接保存	已执行	测试通过	
Test Case 057	回车验证	已执行	测试通过	
Test Case 058	重名校验	已执行	测试通过	
Test Case 059	删除未被使用的类别	已执行	测试通过	
Test Case 060	删除已被使用的类别	已执行	测试通过	
Test Case 061	查看信息显示是否完整	已执行	测试通过	
Test Case 062	翻页	已执行	测试通过	
Test Case 063	必添项是否允许为空	已执行	测试通过	商品管理
Test Case 064	输入字符数等于域允许的最大字符数	已执行	测试通过	
Test Case 065	输入字符数大于域允许的最大字符数	已执行	测试通过	
Test Case 066	回车验证	已执行	测试通过	
Test Case 067	验证系统定义的域长度是否够用	已执行	测试通过	
Test Case 068	重名校验	已执行	测试通过	
Test Case 069	必添项是否允许为空	已执行	测试通过	
Test Case 070	输入字符数等于域允许的最大字符数	已执行	测试通过	
Test Case 071	输入字符数大于域允许的最大字符数	已执行	测试通过	
Test Case 072	不修改直接保存	已执行	测试通过	

续表

测试用例标识号	测试用例名称	用 例 状 态	测 试 结 果	备　　注
Test Case 073	回车验证	已执行	测试通过	
Test Case 074	重名校验	已执行	测试通过	
Test Case 075	删除未被使用的商品	已执行	测试通过	
Test Case 076	删除已被使用的类别	已执行	测试通过	
Test Case 077	查看信息显示是否完整	已执行	测试通过	
Test Case 078	翻页	已执行	测试通过	
购物管理部分				
Test Case 079	翻页	已执行	测试通过	商品查看
Test Case 080	不输入查询	已执行	测试通过	
Test Case 081	选择特定类别查询	已执行	测试通过	
Test Case 082	模糊查询	已执行	测试通过	
Test Case 083	在查询条件中输入特殊字符	已执行	测试通过	
Test Case 084	购买商品	已执行	测试通过	购买商品
Test Case 085	不输入购买数量［购买］	已执行	测试通过	
Test Case 086	在购买数量中输入字母	已执行	测试通过	
Test Case 087	查看购物车	已执行	测试通过	
Test Case 088	生成购物单	已执行	测试通过	
Test Case 089	清空购物车	已执行	测试通过	
Test Case 090	删除商品	已执行	测试通过	
Test Case 091	打印订单	已执行	测试通过	
订单管理部分				
Test Case 092	按时间段查询	已执行	测试通过	订单查询
Test Case 093	起始时间大于结束时间	已执行	测试通过	
Test Case 094	翻页	已执行	测试通过	订单查看
Test Case 095	查看订单详情	已执行	测试通过	订单详情
Test Case 096	确认发货	已执行	测试通过	
Test Case 097	打印发货详单	已执行	测试通过	
系统测试部分				
Test Case 098	大数据量测试	已执行	测试通过	性能测试
Test Case 099	负载测试	已执行	测试通过	
Test Case 100	疲劳强度测试	已执行	测试通过	

续表

测试用例标识号	测试用例名称	用例状态	测试结果	备注
Test Case 101	压力测试	已执行	测试通过	
Test Case 102	按钮状态是否正确	已执行	测试通过	
Test Case 103	按钮的摆放位置是否合理	已执行	测试通过	
Test Case 104	重要按钮的摆放位置是否合适	已执行	测试通过	
Test Case 105	关闭错误提示后的光标定位	已执行	测试通过	
Test Case 106	非法访问	已执行	测试通过	
Test Case 107	所有链接均链接到了该链接的页面	已执行	测试通过	链接测试
Test Case 108	链接的页面不存在	已执行	测试通过	
Test Case 109	系统上没有孤立的页面	已执行	测试通过	
Test Case 110	导航直观	已执行	测试通过	导航
Test Case 111	主页中是否提供了主要模块的链接	已执行	测试通过	
Test Case 112	是否有导航帮助功能	已执行	测试通过	
Test Case 113	导航能否流动到目的地	已执行	测试通过	
Test Case 114	调整浏览器大小，页面还能完全显示	已执行	测试通过	界面测试
Test Case 115	提示、警告、或错误说明应该清楚、明了、恰当	已执行	测试通过	
Test Case 116	是否有错误提示	已执行	测试通过	
Test Case 117	是否有提示说明	已执行	测试通过	
Test Case 118	是否提供放弃的选择项	已执行	测试通过	
Test Case 119	所有页面字体的风格一致	已执行	测试通过	
Test Case 120	背景颜色与字体颜色和前景颜色相搭配	已执行	测试通过	
Test Case 121	表格里的文字折行显示	已执行	测试通过	
Test Case 122	窗体布局	已执行	测试通过	
Test Case 123	页面中的说明文字	已执行	测试通过	
Test Case 124	检查拼写错误	已执行	测试通过	
Test Case 125	分辨率测试	已执行	测试通过	
Test Case 126	浏览器测试	已执行	测试通过	

续表

测试用例 标识号	测试用例 名称	用例状态	测试结果	备注
Test Case 127	平台测试	已执行	测试通过	兼容性测试
Test Case 128	打印机	已执行	测试通过	
Test Case 129	帮助文档中的性能介绍与说明要与系统性能配套一致	已执行	测试通过	
Test Case 130	打包新系统时，对做了修改的地方在帮助文档中要做相应的修改	已执行	测试通过	
Test Case 131	提供搜索功能	已执行	测试通过	帮助文档测试
Test Case 132	提供技术支持方式	已执行	测试通过	
Test Case 133	提供留言功能	已执行	测试通过	
Test Case 134	返回	已执行	测试通过	
Test Case 135	已查看过的商品是否被突出显示	已执行	测试通过	

3.2 测试问题解决

下表中描述测试中发现的、没有满足需求或其他方面要求的部分。

测试用例 标识号	错误或问题 描述	错误或问题 状态
Test Case 001	用户登录系统，未输入用户名的情况下，系统没有给出诸如 “请输入用户名”的提示	已解决
Test Case 081	管理用户查询商品页面，缺少按商品种类字段作为查询条件	已解决
Test Case 085	用户不输入购买数量直接购买商品时没有给出错误提示，用户单击［确定］按钮后，向购物车中添加了该商品，不应该添加	已解决
Test Case 091	用户打印订单，订单从打印机中打出，订单页面布局有错位的地方，不够合理	已解决
Test Case 110	导航按钮不直观，含义不明确	已解决

3.3 测试结果分析

3.3.1 覆盖分析

3.3.1.1 测试覆盖分析

测试覆盖率＝130/135 ×100%=96.29%

需求/功能	用例个数	执行总数	未执行	未/漏测分析和原因
系统功能	109	109	0	产生失败数为 5，最后均以合理的处理方式解决
系统性能	9	9	0	
用户界面	13	13	0	
运行环境	4	4	0	

3.1.1.2　需求覆盖分析

对应约定的测试文档（《网上购物系统测试方案》、《网上购物测试计划》），本次测试对系统需求的覆盖情况为：

需求覆盖率=Y（P）项/需求项总数×100%=83.33%

需求项	测试类型	是否通过[Y][P][N][N/A]	备注
用户手册等	用户测试	[N]	缺少完整的系统安装部署、使用、系统卸载的说明
系统功能	系统测试	[Y]	
系统安全分析	系统测试	[P]	
系统性能	系统测试	[P]	
用户界面	系统测试	[N/A]	
运行环境	系统测试	[P]	

注：P 表示部分通过，N/A 表示不可测试或者用例不适用。

3.3.2　缺陷分析

本次测试中发现 Bug 共 113 个，按严重程度，大部分缺陷级别集中在 B、C 级，即功能性一般缺陷相对较多。

按缺陷在各功能点的分布情况分：

严重级别 / 需求	A-严重影响系统运行的错误	B-功能方面一般缺陷，影响系统运行	C-不影响运行但必须修改	D-合理化建议	<total>
登录系统	1	3	4	2	10
订单详情		2		1	3
修改密码		1			1
修改注册信息		2			2
用户管理	2	4	3		9
购物车管理	1	2	3	1	7
更改密码		3			3
浏览购买记录		1			1
浏览商品		1			1
浏览新闻信息		1			1
找回密码	2				2
注册新用户	2	5	3	1	11
合理性			2	5	7

续表

严重级别 需求	A-严重影响系统运行的错误	B-功能方面一般缺陷，影响系统运行	C-不影响运行但必须修改	D-合理化建议	<total>
易用性		1	1		2
友好性			1	1	2
一致性			2	1	3
正确性	1				1
系统安全分析	1	1	1		3
发布公告信息	1	1	1		3
查询用户信息		2	4		6
填写用户信息		1	1		2
修改商品信息	2	14	9	1	26
<total>	13	45	35	13	106

可以看出：缺陷大部分集中在修改商品信息、注册新用户以及登录系统部分，其余分布较为分散。

四、综合评价

4.1　软件能力

经过项目组开发人员、测试组人员以及相关人员的协力合作，网上购物系统项目如期交付并达到交付标准。该系统能够实现网上购物系统项目在用户需求说明书中所约定的功能，即能够基本满足用户在前台进行用户注册，登录，购买物品以及搜索和浏览商品列表信息，商家在网上购物系统后台可根据自己公司需要更改商品信息。

4.2　缺陷和限制

该系统除基本满足功能需求外，在性能方面还存在不足，有系统继续优化的空间。另外，部分功能在设计上仍存在不足之处。

4.3　建议

需求提出方可以在使用该系统的基础上，继续搜集用户的使用需求反馈，并结合市场同类产品的优势，在今后的版本中不断补充并完善功能。

另外，建议当项目组成员确定后，在项目组内部对一些事项进行约定。如 Web 开发/测试的通用规范等，将会在一定程度上提高开发和测试的效率。

4.1.2　应知应会

1．测试总结与测试报告

对软件进行测试是为了检查软件的合理性，尽量多地发现软件的缺陷。测试总结则是用于分析并总结这些缺陷，将总结出来的经验用于指导下一次的测试用例设计。在每个版本测试完毕后，要进行测试总结，做一些比例的总结、缺陷严重级别及比例的总结，人员工作效率的总结等等，最重要的是风险的评估，还有对下一测试版本的建议等。

测试报告按标准流程是需要每天写的，其中包括执行的测试用例，发现的缺陷有哪些，这些缺陷的详细情况等。

2．各种模板

下面将测试报告模板、测试总结模板、系统验收测试分析报告模板附在后面，供读者参考。

测试报告（模板）

一、引言

1.1　编写目的

说明这份测试分析报告的具体编写目的，指出预期的阅读范围。

1.2　背景

说明：

被测试软件系统的名称；

该软件的任务提出者、开发者、用户及安装此软件的计算中心，指出测试环境与实际运行环境之间可能存在的差异以及这些差异对测试结果的影响。

1.3　定义

列出本文件中用到的专用术语的定义和外文首字母词组的原词组。

1.4　参考资料

列出要用到的参考资料，如：

本项目的经核准的计划任务书或合同、上级机关的批文；

属于本项目的其他已发表的文件；

本文件中各处引用的文件、资料，包括所要用到的软件开发标准。列出这些文件的标题、文件编号、发表日期和出版单位，说明足够得到这些文件资料的来源。

二、测试概要

用表格的形式列出每一项测试的标识符及其测试内容，并指明实际进行的测试工作内容与测试计划中预先设计的内容之间的差别，说明做出这种改变的原因。

三、测试结果及发现

测试 1（标识符）

把本项测试中实际得到的动态输出（包括内部生成数据输出）结果同对于动态输出的要求进行比较，陈述其中的各项发现。

<table>
<tr><td>设计人</td><td colspan="2"></td><td>测试人</td><td></td><td>功能编号</td><td>（按顺序自行编号）</td></tr>
<tr><td>功能组</td><td colspan="2">（如主页、政务公开频道）</td><td>功能点</td><td>（如用户登录）</td><td>测试日期</td><td></td></tr>
<tr><td colspan="7">测 试 环 境 及 前 提</td></tr>
<tr><td>测试 URL</td><td colspan="6"></td></tr>
<tr><td>测试条件</td><td colspan="6">（如：已注册用户。如果用户没有注册，先进行用户注册）</td></tr>
<tr><td colspan="7">测 试 项 目 及 内 容</td></tr>
<tr><td>测试步骤</td><td colspan="2">输入项</td><td colspan="2">预期输出项</td><td colspan="2">实际输出</td></tr>
<tr><td rowspan="2">1</td><td colspan="2" rowspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td rowspan="3">2</td><td colspan="2" rowspan="3"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>3</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="7">测 试 结 论</td></tr>
<tr><td>测试记录</td><td colspan="3"></td><td>总体结论</td><td colspan="2">□通过　□基本通过　□未通过</td></tr>
</table>

用类似本报告 3.1 条的方式给出第 2 项及其后各项测试内容的测试结果和发现。

四、对软件功能的结论

4.1　功能 1（标识符）

4.1.1　能力

简述该项功能，说明为满足此项功能而设计的软件能力以及经过一项或多项测试已证实的能力。

4.1.2　限制

说明测试数据值的范围（包括动态数据和静态数据），列出就这项功能而言，测试期间在该软件中查出的缺陷、局限性。

4.2　功能 2（标识符）

用类似本报告 4.l 的方式给出第 2 项及其后各项功能的测试结论。

……

五、分析摘要

5.1　能力

陈述经测试证实了的本软件的能力。如果所进行的测试是为了验证一项或几项特定性能要求的实现，应提供这方面的测试结果与要求之间的比较，并确定测试环境与实际运行环境之间可能存在的差异对能力的测试所带来的影响。

5.2　缺陷和限制

陈述经测试证实的软件缺陷和限制，说明每项缺陷和限制对软件性能的影响，并说明全部测得的性能缺陷的累积影响和总影响。

5.3　建议

对每项缺陷提出改进建议，如：

各项修改可采用的修改方法；

各项修改的紧迫程度；

各项修改预计的工作量；

各项修改的负责人。

5.4　评价

说明该项软件的开发是否已达到预定目标，能否交付使用。

六、测试资源消耗

总结测试工作的资源消耗数据，如工作人员的水平级别数量、机时消耗等。

测试总结编写指南

摘要

测试总结是把测试的过程和结果写成文档，并对发现的问题和缺陷进行分析，为纠正软件的存在的质量问题提供依据，同时为软件验收和交付打下基础。本文提供测试总结模板以及如何编写的实例指南。

关键字

测试总结　缺陷

正文

测试总结是测试阶段最后的文档产物，优秀的测试经理应该具备良好的文档编写能力，一份详细的测试总结包含足够的信息，包括产品质量和测试过程的评价，测试总结基于测试

中的数据采集以及对最终的测试结果分析。

下面以通用的测试总结模板为例，详细展开对测试总结编写的具体描述。

PARTⅠ　首页

0.1　页面内容：

密级

通常，测试总结供内部测试完毕后使用，因此密级为中，如果可供用户和更多的人阅读，密级为低，高密级的测试总结适合内部研发项目以及涉及保密行业和技术版权的项目。

××××项目/系统测试总结

总结编号

（可供索引的内部编号或者用户要求分布提交时的序列号）

部门经理 ______项目经理______

开发经理______测试经理______

×××公司 ××××单位 （此处包含用户单位以及研发此系统的公司）

××××年××月××日

0.2　格式要求：

标题一般采用大字体（如一号），加粗，宋体，居中排列；

副标题采用小一号字（如二号）加粗，宋体，居中排列；

其他采用四号字，宋体，居中排列。

0.3　版本控制：

版本 作者 时间 变更摘要；

新建/变更/审核。

PARTⅡ 引言部分

1.1　编写目的

本测试总结了具体编写目的，指出预期的读者范围。

实例：本测试总结为×××项目的测试总结，目的在于总结测试阶段的测试以及分析测试结果，描述系统是否符合需求（或达到×××功能目标）。预期参考人员包括用户、测试人员、开发人员、项目管理者、其他质量管理人员和需要阅读本总结的高层经理。

提示：通常，用户对测试结论部分感兴趣，开发人员希望从缺陷结果以及分析得到产品开发质量的信息，项目管理者对测试执行中成本、资源和时间予与重视，而高层经理希望能够阅读到简单的图表并且能够与其他项目进行同向比较。此部分可以具体描述为什么类型的人可参考本总结××页××章节，你的总结读者越多，你的工作越容易被人重视，前提是必须让阅读者感到你的总结是有价值而且值得浪费一点时间去关注的。

1.2　项目背景

对项目目标和目的进行简要说明。必要时包括简史，这部分不需要脑力劳动，直接从需求或者招标文件中复制即可。

1.3　系统简介

如果设计说明书有此部分，照抄。注意必要的框架图和网络拓扑图能吸引眼球。

1.4　术语和缩写词

列出设计本系统/项目的专用术语和缩写语约定。对于技术相关的名词和多义词一定要标注清楚，以便阅读时不会产生歧义。

1.5　参考资料

1．需求、设计、测试用例、手册以及其他项目文档都是范围内可参考的东西。

2．测试使用的国家标准、行业指标、公司规范和质量手册等。

PART Ⅲ 测试概要

测试的概要介绍，包括测试的一些声明、测试范围、测试目的等，主要是测试情况简介（其他测试经理和质量人员关注部分）。

2.1　测试用例设计

简要介绍测试用例的设计方法。例如：等价类划分、边界值、因果图，以及用这类方法进行的设计（3～4 句）。

提示：如果能够具体对设计进行说明，在其他开发人员、测试经理阅读的时候就容易对你的用例设计有个整体的概念，顺便说一句，在这里写上一些非常规的设计方法也是有利的，至少在没有看到测试结论之前就可以了解到测试经理的设计技术，重点测试部分一定要保证有两种以上不同的用例设计方法。

2.2　测试环境与配置

本部分简要介绍测试环境及其配置。

提示：清单如下，如果系统/项目比较大，则用表格方式列出。

- 数据库服务器配置
- CPU
- 内存
- 硬盘：可用空间大小
- 操作系统
- 应用软件
- 机器网络名
- 局域网地址
- 应用服务器配置
- 客户端配置
- ……

对于网络设备和要求也可以使用相应的表格，对于三层架构的，可以根据网络拓扑图列出相关配置。

2.3　测试方法（和工具）

简要介绍测试中采用的方法（和工具）。

提示：主要是黑盒测试，测试方法可以写上测试的重点和采用的测试模式，这样可以一目了然地知道是否遗漏了重要的测试点和关键块。工具为可选项，当使用到测试工具和相关工具时，要说明。注意要注明是自产还是厂商，版本号多少，在测试总结发布后要避免许多工具的版权问题。

PART Ⅳ 测试结果及缺陷分析

整个测试总结中这是最激动人心的部分，这部分主要汇总各种数据并进行度量，度量包括对测试过程的度量和能力评估、对软件产品的质量度量和产品评估。对于不需要过程度量或者相对较小的项目，例如用于验收时提交用户的测试总结、小型项目的测试总结，可省略过程方面的度量部分；而采用了 CMM/ISO 或者其他工程标准过程的，需要提供过程改进建

议和参考的测试总结－主要用于公司内部测试改进和缺陷预防机制－则过程度量需要列出。

3.1　测试执行情况与记录

描述测试资源消耗情况，记录实际数据（测试、项目经理关注部分）。

3.1.1　测试组织

可列出简单的测试组架构图，包括：

- 测试组架构（如存在分组、用户参与等情况）
- 测试经理（领导人员）
- 主要测试人员
- 参与测试人员

3.1.2　测试时间

列出测试的跨度和工作量，最好区分测试文档和活动的时间。数据可供过程度量使用。

例如，×××子系统/子功能要列出的内容如下：

- 实际开始时间－实际结束时间
- 总工时/总工作日
- 任务开始时间和结束时间
- 时间的总计或合计

对于大系统/项目来说最终要统计资源的总投入，必要时要增加成本一栏，以便管理者清楚地知道究竟花费了多少人力去完成测试。成本中可以列出如下内容：

- 测试类型
- 人员成本
- 工具设备
- 其他费用
- 费用总计

在数据汇总时可以统计个人的平均投入时间和总体时间、整体投入平均时间和总体时间，还可以算出每一个功能点所花费的时/人。具体内容如下：

- 用时人员
- 编写用例
- 执行测试
- 总计

另外还要列出用于过程度量的数据包括文档生产率和测试执行率。内容如下：

- 生产率人员
- 用例/编写时间
- 用例/执行时间
- 平均
- 合计

3.1.3　测试版本

给出测试的版本，如果是最终总结，可能要总结测试次数回归测试多少次。列出表格清单则便于知道那个子系统/子模块的测试频度，对于多次回归的子系统/子模块将引起开发者关注。

3.2　覆盖分析

3.2.1　需求覆盖

需求覆盖率是指经过测试的需求/功能和需求规格说明书中所有需求/功能的比值，通常情况下要达到100%的目标。具体列出如下内容：

- 需求/功能（或编号）
- 测试类型
- 是否通过
- 备注

根据测试结果，按编号给出每一测试需求通过与否的结论。可以用P表示部分通过，N/A表示不可测试或者用例不适用。实际上，需求跟踪矩阵列出了一一对应的用例情况以避免遗漏，作用为传达需求的测试信息以供检查和审核。

需求覆盖率计算公式如下：

需求覆盖率=通过测试的项数/需求总数 ×100%

3.2.2　测试覆盖

此部分要分析测试的覆盖率，要列出以下内容：

- 需求/功能（或编号）
- 用例个数
- 执行总数
- 未执行总数
- 未/漏测分析和原因

实际上，测试用例已经记载了预期结果数据，测试缺陷上说明了实测结果数据和与预期结果数据的偏差；因此没有必要对每个编号在此包含更详细说明的缺陷记录与偏差，列表的目的仅在于更好地查看测试结果。

测试覆盖率计算公式如下：

测试覆盖率=执行数/用例总数 ×100%

3.3　缺陷的统计与分析

缺陷统计主要涉及到被测系统的质量，因此，这部分成为开发人员、质量人员重点关注的部分。

3.3.1　缺陷汇总

本部分将缺陷进行汇总。

按所测试的阶段列出如下内容：

- 被测试系统中缺陷的严重程度
- 系统测试中缺陷的严重程度
- 回归测试中缺陷的严重程度

严重程度可以分为严重、一般、微小。

按缺陷类型列出如下内容：

- 用户界面
- 一致性
- 功能
- 算法

- 接口
- 文档
- 用户界面
- 其他

按功能分布列出如下内容：

- 功能一
- 功能二
- 功能三
- 功能四
- 功能五
- 功能六
- 功能七
- ……

最好给出缺陷的饼状图和柱状图以便直观查看。俗话说一图胜千言，图标能够使阅读者迅速获得信息，尤其是各层面管理人员没有时间去逐项阅读文章。

3.3.2　缺陷分析

本部分是对上述缺陷和其他收集数据进行综合分析。

缺陷综合分析公式如下：

缺陷发现效率＝缺陷总数/执行测试用时

可到具体人员得出平均指标，公式如下：

用例质量＝缺陷总数/测试用例总数 ×100%

缺陷密度＝缺陷总数/功能点总数

缺陷密度可以得出系统各功能或各需求的缺陷分布情况，开发人员可以在此分析基础上可以得出哪部分功能或者需求的缺陷最多，从而在今后开发注意避免并注意在实施时予与关注，测试经验表明，测试缺陷越多的部分，其隐藏的缺陷也越多。

并且在此部分还要附上测试曲线图，描绘被测系统每工作日或每周的缺陷数情况，得出缺陷走势和趋向。

最后还要以表格的形式列出重要缺陷摘要，包括：缺陷编号、简要描述、分析结果等。

3.3.3　残留缺陷与未解决问题

本部分列出测试之后尚残留的缺陷以及未解决的问题。

可将残留缺陷归纳成如下项：

- 残留缺陷编号
- BUG 号
- 缺陷概要：指该缺陷描述的事实
- 原因分析：指如何引起缺陷，缺陷的后果，描述造成软件局限性和其他限制性的原因
- 预防和改进措施：指弥补手段和长期策略
- 未解决问题
- 功能/测试类型
- 测试结果：指与预期结果的偏差
- 缺陷：指缺陷的具体描述

● 评价：是对这些问题的看法，也就是这些问题如果发出去了会造成什么样的影响

PART Ⅴ 测试结论与建议

总结到了这个部分就是一个总结了，对上述过程、缺陷分析之后该下个结论，此部分为项目经理、部门经理以及高层经理关注，请清晰扼要的下定论。

4.1 测试结论

1．测试执行是否充分（可以增加对安全性、可靠性、可维护性和功能性描述）

2．对测试风险的控制措施和成效

3．测试目标是否完成

4．测试是否通过

5．是否可以进入下一阶段项目目标

4.2 建议

1．对系统存在问题的说明，描述测试所揭露的软件缺陷和不足，以及可能给软件实施和运行带来的影响

2．可能存在的潜在缺陷和后续工作

3．对缺陷修改和产品设计的建议

4．对过程改进方面的建议

测试总结的内容大同小异，对于一些测试总结而言，可能将第四和第五部分合并，逐项列出测试项、缺陷、分析和建议，这种方法也比较多见，尤其在第三方评测总结中，此份总结模板仅供参考。

系统验收测试分析报告

（参考模板）

一、概述

1.1　编写目的

编写测试分析报告是为达到以下目的：

1．对系统测试工作进行总结；

2．对测试的各个阶段进行评价，并对测试结果进行分析；

3．为纠正软件缺陷提供依据。

1.2　参考资料

列出系统测试所参考的资料，需求分析、系统设计、用户手册等。

1.3　术语和缩写词

说明本次测试所涉及到的专业术语和缩写词等。

1.4　测试人员安排和分工

列出本次系统测试实际的人员分工。

人员	单位	职务	分工
		测试负责人	
		测试员	
		测试员	

1.5　测试环境

列出系统验收测试使用的软硬件环境。

1.6　测试工作流程

列出系统测试的工作流程。

二、测试内容

根据测试计划中编写的测试用例，列出系统验收测试的测试内容。

三、测试结果分析

汇总测试发现的问题，通过对测试结果的分析提出一个对软件能力的全面分析，需标明遗留缺陷、局限性和软件的约束限制等，并提出改进建议。

四、测试总结

对整个系统测试过程和测试系统的质量做总结评价，根据测试标准及测试结果，判定软件是否通过测试。

4.1.3　拓展任务

独立编写天天超市管理系统的测试报告及测试总结。

第 5 章　测 试 工 具

本章介绍 LoadRunner 8.0 的安装过程，测试脚本的录制，脚本的调试与完善。

本章重点：

- LoadRunner 8.0 脚本的录制
- LoadRunner 8.0 脚本的调试与完善

LoadRunner 是一种适用于各种体系架构的自动负载测试工具，它能预测系统行为并优化系统性能。LoadRunner 的测试对象是整个企业的系统，它通过模拟实际用户的操作行为和实行实时性能监测，来帮助你更快地查找和发现问题。此外，LoadRunner 能支持广范的协议和技术，为你的特殊环境提供特殊的解决方案。

工作任务 5.1　LoadRunner 8.0 的安装过程

5.1.1　LoadRunner 8.0 的安装过程

1．工作任务描述

LoadRunner 分为 Windows 版和 UNIX 版，由于我们的测试实例是在 Windows 环境下进行的，因此本节只介绍 Windows 版 LoadRunner 的安装。安装建议说明：

（1）最好将 LoadRunner 8.0 安装到系统的根目录上。例如：c:\loadrunner8.0

（2）尽量保证要安装此系统计算机的干净的环境。

（3）安装时的最低配置要求 Windwos 2000 或者更高。

（4）管理中心的最低配置要求 Windows 2000 Server（好像不支持 Windows Professional）

（5）一定要以 Administrator 身份安装！

（6）安装的时候关闭所有的杀毒软件！

2．工作过程

（1）系统要求

要比较好的运行 LoadRunner，要求机器内存在 512MB 以上，安装 LoadRunner 的磁盘空间至少剩余 1GB，操作系统最好为 Windows 2000 或 Windows 2003。

（2）安装过程

① 以 Administrator 的身份登录 Windows 2003 后，运行 LoadRunner 安装目录中。

Setup.exe 即可进入安装程序。如图 5-1 所示，选择第一项 LoadRunner Full Setup，完全安装，将出现 License Agreement 界面，如图 5-2 所示。

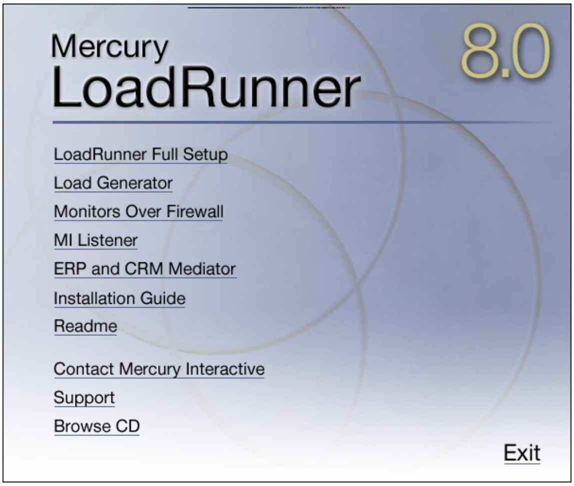


图 5-1　LoadRunner 安装目录界面

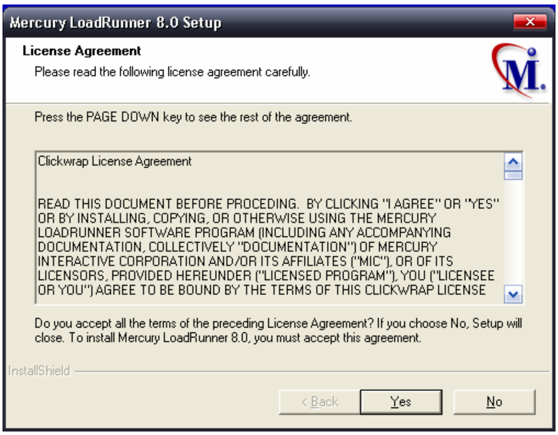


图 5-2　许可协议界面

② 在此界面中描述了 Mercury Interactive 公司的版权声明，接受声明中的许可协议则可单击［Yes］按钮，进入下一个“Setup Type”界面，如图 5-3 所示。

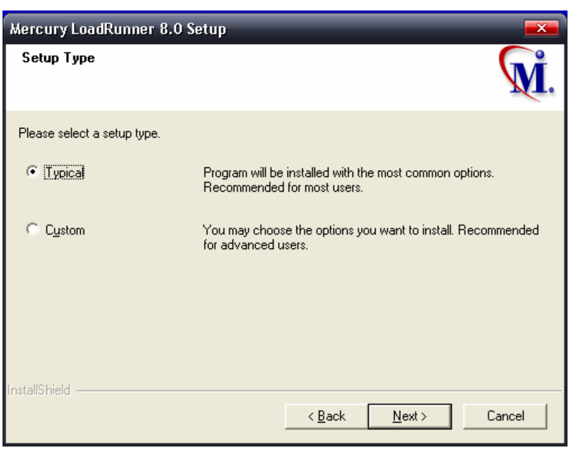


图 5-3　安装类型界面

③ 在“Setup Type”界面中，需要选择一种安装类型，建议选择“Typical”典型安装，这样可以一次性安装所有组件。否则可以选择“Custom”即自定义安装，可以自选要安装的组件。选择了安装类型后，单击［Next］按钮，将进入“License Information”界面，如图 5-4 所示。

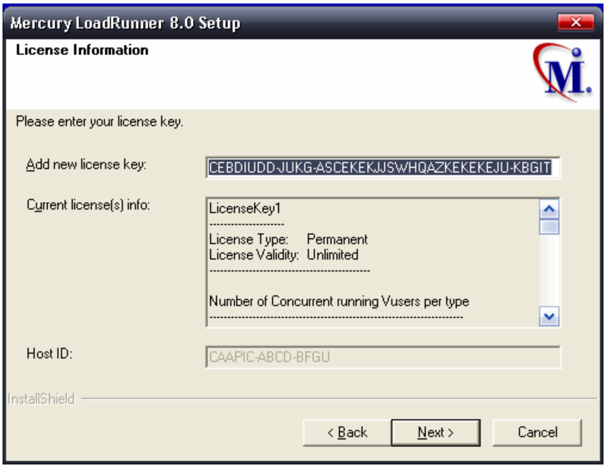


图 5-4　安装许可信息界面

④ 在“License Information”界面，输入 License Key，单击［Next］按钮继续，进入“Choose Destination Location”界面，如图 5-5 所示。

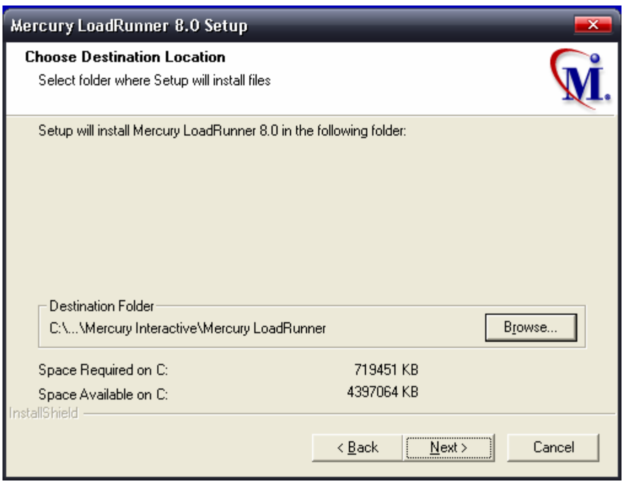


图 5-5　选择安装位置界面

⑤ 在“Choose Destination Location”界面，如果是网络安装，则要安装到一个网络驱动器上。提醒：最好把网络驱动器映射成本机的一个盘符（比如 H:盘），安装 LoadRunner 的各级目录名称不要包含中文字符。设置好安装目录后，单击［Next］将进入“Web Server Username”界面，如图 5-6 所示。

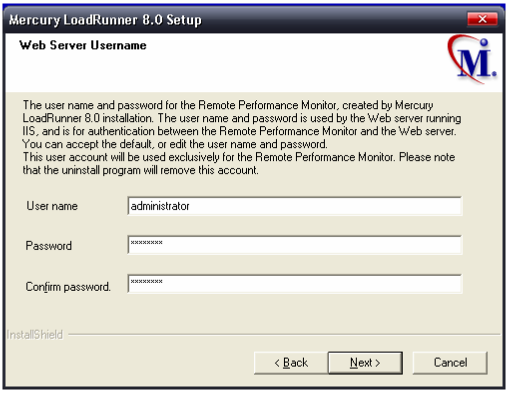


图 5-6　Web 服务器信息设置界面

⑥ 在“Web Server Username”界面，需要设置一组用户名及口令。这组用户名及口令将用于远程性能监控终端对 Web 服务器进行远程监控时的身份验证。设置用户名及口令后，单击［Next］，将进入“Select Program Folder”界面，如图 5-7 所示。

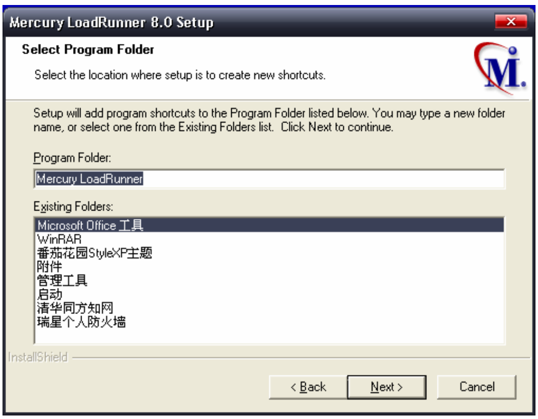


图 5-7　设置程序目录界面

⑦ 在“Select Program Folder”界面，需要设置程序目录中当前程序的快捷方式，单击［Next］按钮，将进入“Review Settings”界面，如图 5-8 所示。

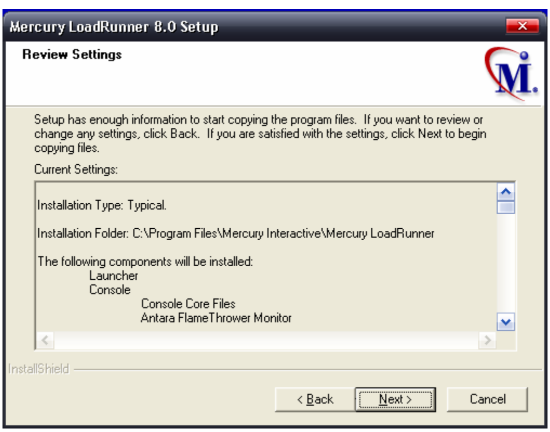


图 5-8　重新检查安装目录界面

在“Review Settings”界面，继续单击［Next］按钮，将进入拷贝文件界面，到此便可等待 LoadRunner 安装成功完成了。

5.1.2　应知应会

1．LoadRunner 8.0 简介

LoadRunner 是一种预测系统行为和性能的工业标准级负载测试工具。通过以模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题，LoadRunner 能够对整个企业架构进行测试。通过使用 LoadRunner，企业能最大限度地缩短测试时间，优化性能和加速应用系统的发布周期。

目前企业的网络应用环境都必须支持大量用户，网络体系架构中含各类应用环境且由不同供应商提供软件和硬件产品。难以预知的用户负载和愈来愈复杂的应用环境使公司时时担心会发生用户响应速度过慢，系统崩溃等问题。这些都不可避免地导致公司收益的损失。

Mercury Interactive 的 LoadRunner 能让企业保护自己的收入来源，无须购置额外硬件而最大限度地利用现有的 IT 资源，并确保终端用户在应用系统的各个环节中对其测试应用的质量，可靠性和可扩展性都有良好的评价。LoadRunner 是一种具备高度适应性的，自动负载测试工具，它能预测系统行为，优化系统性能。LoadRunner 强调的是整个企业系统，它通过模拟实际用户的操作行为和实施实时性能检测，来帮助你更快地确认和查找问题的所在。此外，LoadRunner 能支持最广泛的协议标准和技术，为你的特殊环境量身定做提供解决方案。

2．性能测试的主要术语

（1）场景

场景是一种文件，用于根据性能要求定义在每一个测试会话运行期间发生的事件。

（2）Vuser

在场景中 LoadRunner 用虚拟用户或 Vuser 代替实际用户。Vuser 模拟实际用户的操作来使用应用程序。

（3）测试脚本

测试脚本用于描述 Vuser 在测试场景中所执行的操作。

（4）并发

狭义的并发一般分两种情况。一种是严格意义上的并发，即所有用户在同一时刻做同一件事情或操作，这种操作一般针对同一类型的业务。

另一种并发是广义的并发。这种并发与狭义的并发的区别是尽管多个用户对系统发出了请求或进行了操作，但是这些请求或操作可以是相同的，也可以是不同的。对整体系统而言，虽然有很多用户同时对系统进行操作，因此，仍然属于并发的范畴。

可以看出，广义的并发是包含狭义的并发的，而且广义的并发更接近用户的实际使用情况，因为对大多数系统而言，只有数量很少的用户进行“严格意义上的并发”。对于性能测试而言，这两种并发一般都需要进行测试，通常的做法是先进行严格意义上的并发测试。严格意义上的并发一般发生在使用比较频繁的模块中，尽管发生的概率不是特别高，但是一旦发生性能问题，后果很可能是致命的。严格意义上的并发测试往往和 功能测试关联起来，因为只要并发功能遇到异常通常都是程序的问题，这种测试也是健壮性和稳定性测试的一部分。

（5）并发用户数量

关于并发用户数量，有两种常见的错误观点。一种错误观点是把并发用户数量理解为使

用系统的全部用户的数量，理由是这些用户可能同时使用系统；还有一种比较接近正确的观点是把用户在线数量理解为并发用户数量。实际上，在线用户不一定会和 其他用户发生并发，例如正在浏览网页信息的用户，对服务器是没有任何影响的。但是，用户在线数量是统计并发用户数量的主要依据之一。

并发主要针对服务器而言，是否并发的关键是看用户的操作是否对服务器产生了影响。因此，并发用户数量的正确理解是，在同一时刻与服务器进行交互的在线用户数量。这些用户的最大特征是和服务器发生了交互，这种交互既可以是单向传送数据的，也可以是双向传送数据的。

并发用户数量的统计方法目前还没有准确的公式，因为不同的系统会有不同的并发特点。例如 OA 系统统计并发用户的经验公式为：使用系统的用户数量×（5%～20%）。对于这个公式，没有必要拘泥于计算出的结果，因为为了保证系统的扩展空间，测试时的并发用户数量就会稍稍大一些，除非要测试系统能承受的最大并发用户数量。举例说明：如果一个 OA 系统的期望用户为 1000 个，只要测试出系统能支持 200 个并发用户就可以了。

（6）请求响应时间

请求响应时间是指从客户端发出请求到得到响应的整个过程的时间。这个过程从客户端发出一个请求开始计时，到客户端接收到从服务器端返回的响应结果计时结束。在某些工具中，请求响应时间通常会被称为“TTLB”，即“Time to last byte”，意思是从发送一个请求开始，到客户端接收到最后一个字节的响应为止所耗费的时间。请求响应时间的单位一般为“秒”或“毫秒”。

（7）事物响应时间

事物可能由一系列请求组成，事物的响应时间主要针对用户而言，属于宏观上的概念，是为了向用户说明业务响应时间而提出来的。例如：跨行取款事物的响应时间就是由一系列的请求组成的。事物响应时间和业务吞吐率都是直接衡量系统性能的参数。

（8）吞吐量

指在一次性能测试过程中网络上传输的数据量的总和。吞吐量/传输时间，就是吞吐率。

（9）吞吐率

通常用来指单位时间内网络上传输的数据量，也可以指单位时间内处理的客户端请求数量，是衡量网络性能的重要指标。

但是从用户或业务角度来看，吞吐率也可以用“请求数/秒”或“页面数/秒”、“业务数/小时或天”、“访问人数/天”、“页面访问量/天”来衡量。例如在银行卡审批系统中，可以用“千件/每小时”来衡量系统的业务处理能力。

（10）TPS

每秒钟系统能够处理的交易或事物的数量。它是衡量系统处理能力的重要指标。TPS是LoadRunner中重要的性能参数指标。

（11）单击率

每秒钟用户向 Web 服务器提交的 HTTP 请求书。这个指标是 Web 应用特有的一个指标：Web 应用是“请求-响应”模式，用户发出一次申请，服务器就要处理一次，所以“单击”是 Web 应用能够处理交易的最小单位。如果把每次单击定义为一次交易，单击率和 TPS 就是一个概念。不难看出，点击率越大，对服务器的压力也越大。单击率只是一个性能参考指标，重要的是分析单击时产生的影响。

需要注意的是，这里的点击不是指鼠标的一次“单击”操作，而是在一次“单击”操作中，客户端可能向服务器发出多个 HTTP 请求。

（12）资源利用率

资源利用率指的是对不同系统资源的使用程度，例如服务器的CPU利用率、磁盘利用率等。资源利用率是分析系统性能指标而改善性能的主要依据，因此，它是Web性能测试 工作的重点。

资源利用率主要针对Web服务器、操作系统、数据库服务器、网络等，是测试和分析瓶颈的主要参数。在性能测试中，要根据需求采集具体的资源利用率参数来进行分析。

3．LoadRunner的License类型

LoadRunner 的 License 根据协议、用户数量和期限有着明显的区分：

（1）按照场景运行协议分为可支持 B/S 结构的系统和 C/S 结构的系统两种。

（2）按 License 可支持的 Vuser——虚拟用户数量上分为，无限制数量和固定数量两种，无限制数量的指虚拟用户数量无穷大，固定数量的即虚拟用户数为具体数值如：10000。

（3）按有实效日期又可分为无限制使用期的和有限使用天数的。

5.1.3　拓展任务

可以深入学习一下 LoadRunner 的一些属性和参数。

工作任务 5.2　LoadRunner 8.0 生成测试脚本

5.2.1　LoadRunner 录制脚本

1．工作任务描述

生成测试脚本可分两步完成，第一步，利用 LoadRunner 的脚本录制工具 Visual User Generator（以下简称 VuGen）录制一个基础的测试脚本，第二步，对录制好的基础脚本，按照测试场景的需要进一步调试及完善，以下为具体操作过程。

2．工作过程

录制基础脚本：

（1）启动 Visual User Generator 后，在主窗口，如图 5-9 所示，单击“File”菜单，选择“New…”菜单项来新建一个测试脚本。

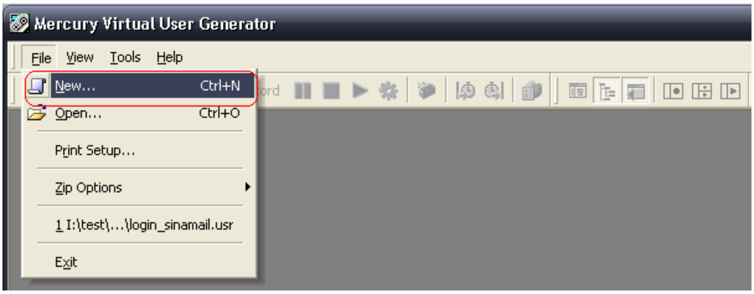


图 5-9　新建脚本菜单

（2）新建测试脚本第一步是选择系统通信的协议，协议是客户端用来与系统后端进行通

信的语言，在“New Virtual User”窗口左侧有三个选项：

- “New Single Protocol Script”，选择此项，则在对话框右侧视图中只能选择一种协议。
- “New Multiple Protocol Script”，选择此项，则在对话框右侧视图中可以根据测试需要选择多种协议。
- “New Script Recent Protocols”，选择此项，则在对话框右侧视图中，可以选择一种最新用过的协议。如图 5-10 所示。

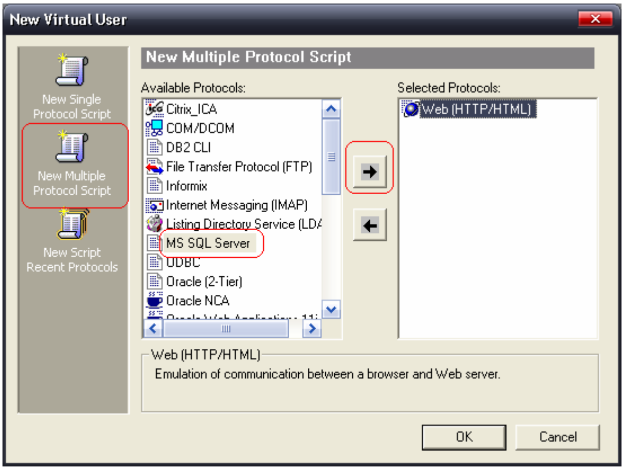


图 5-10　选择系统通信协议对话框

（3）在“New Virtual User”对话框单击［OK］按钮，将进入“Start Recording”对话框，如果在上一步选择的是“New Multiple Protocol Script”按钮，则在“Start Recording”对话框中需要设置如下项目：

- “Application type”　选择 Internet Applications
- “Programe to record”选择 Microsoft Internet Explorer
- “URL Address”输入被测试系统的访问 URL
- “Working Directory”输入运行测试脚本的路径
- “Record into Action”建议选择 Action，也可以单击［New…］按钮来新建一个函数名称。

如果在上一步选择了“New Single Protocol Script”，则进入“Start Recording”对话框后，只输入“URL：”和“Record into Action：”即可，如图 5-11 所示。

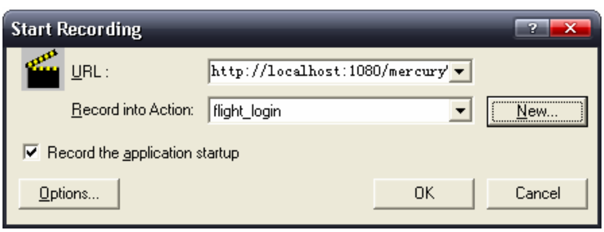


图 5-11　录制配置对话框

以 LoadRunner 自带的 Mercury Tours 为例，在“URL”地址框中输入 http://localhost:1080/mercuryWebTours，单击“Record into Action”后面的［New］按钮，输入新的函数名称，单

击［OK］按钮，便开始录制测试脚本。

（4）“Record the application startup”默认情况下是选中的，说明应用程序一旦启动，VuGen 就会开始录制脚本；如果没有选中，应用程序启动后，VuGen 出现以下对话框，并且暂时不会开始录制脚本，用户操作应用程序到需要录制的地方，按下［Record］按钮，VuGen 才开始录制。如图 5-12 所示。



图 5-12　设置录制窗口

（5）单击［Options］按钮，进入录制的设置窗体，这里一般情况下不需要改动。如图 5-13 所示。

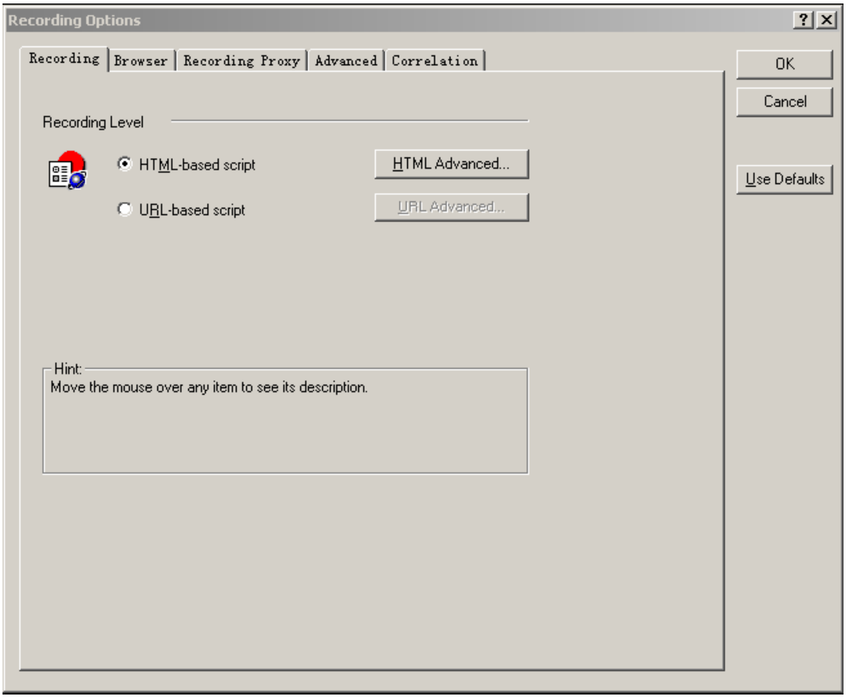


图 5-13　环境设置界面

➢ Recording 标签页：

① 默认情况下选择“HTML-based Script”，说明脚本中采用 HTML 页面的形式来表示，这种方式的 Script 脚本容易维护，容易理解，推荐这种方式录制。

② “URL-based Script”说明脚本中的表示采用基于 URL 的方式，这种方式看上去比较乱。选择哪种方式录制，有以下参考原则：

基于浏览器的应用程序推荐使用 HTML-based Script。

不是基于浏览器的应用程序推荐使用 URL-based Script。

如果基于浏览器的应用程序中包含了 JavaScript 并且该脚本向服务器产生了请求，基于浏览器的应用程序中使用了 HTTPS 安全协议，使用 URL-based 方式录制。

➢ Advanced 标签页：取默认情况即可。下面的图简单的说明了各项的含义。如图 5-14 所示。

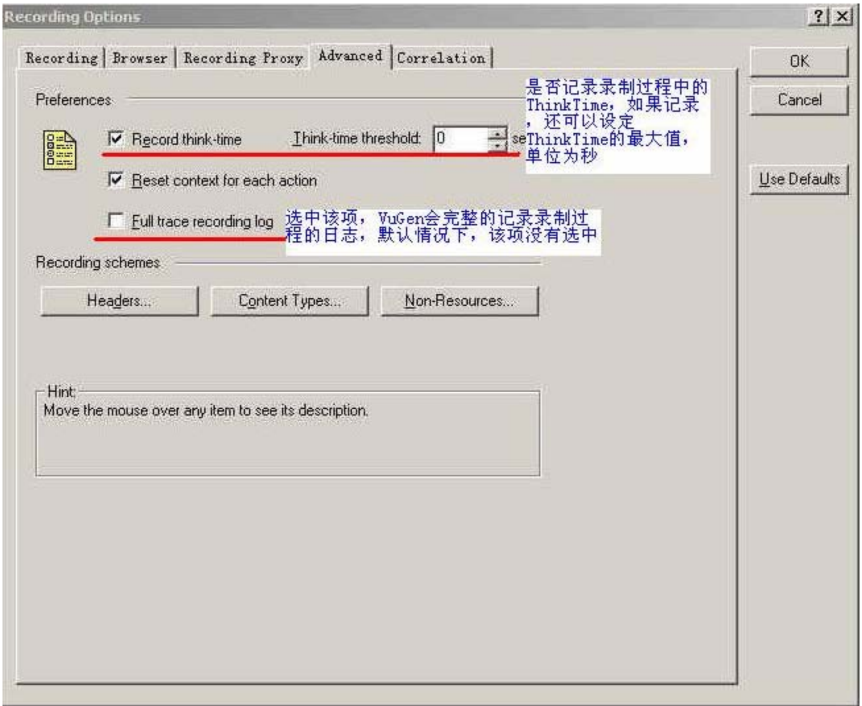


图 5-14　Advanced 标签页

➢ Correlation 标签页：主要是为了在录制过程中设置自动关联。（没有环境，不做详细描述）然后单击［OK］按钮后，VuGen 开始录制脚本。

在录制过程中，不要使用浏览器的“后退”功能，容易出错！

录制过程中，在屏幕上会有一个工具条出现。下面我们简单介绍一下各个按钮的功能。如图 5-15 所示。

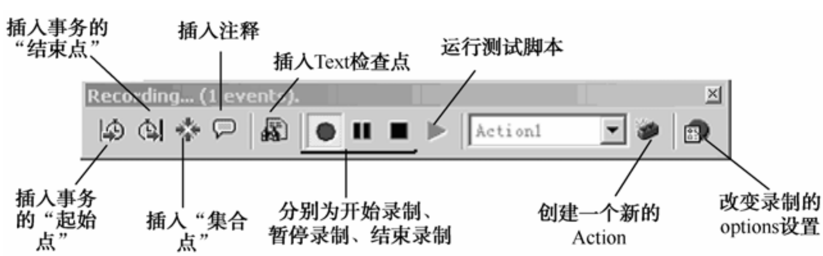


图 5-15　录制过程工具条

（6）首先，录制脚本工具 VuGen 将打开一个新的 Web 浏览器，并访问到“URL Address”输入的地址，即 Mercury Tours 主页面。

（7）接下来，就可以根据功能测试操作步骤来操作被测试的应用系统。

（8）操作结束后，单击 Visual User Generator 工具中的［录制结束］按钮，VuGen 将自动生成测试脚本，如图 5-16 所示，并退出录制过程。

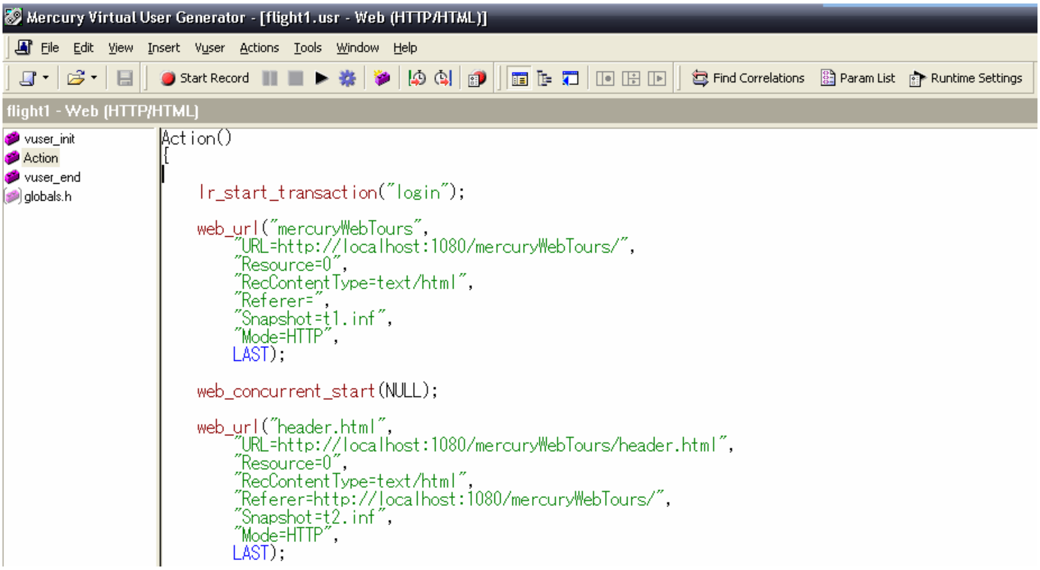


图 5-16　脚本图例窗口

注意事项：

（1）VuGen 中的脚本分为三部分：vuser_init、vuser_end 和 Action。其中 vuser_init 和 vuser_end 都只能存在一个，不能再分割，而 Action 还可以分成无数多个部分（通过单击［New］按钮，新建 ActionXXX）。

（2）通常，在录制需要登录的系统时，我们把登录部分放到 vuser_init 中，把登录后的操作部分放到 Action 中，把注销关闭登录部分放到 vuser_end 中。（如果需要在登录操作设集合点，那么登录操作也要放到 Action 中，因为 vuser_init 中不能添加集合点）在其他情况下，我们只要把操作部分放到 Action 中即可。

（3）在重复执行测试脚本时，vuser_init 和 vuser_end 中的内容只会执行一次，重复执行的只是 Action 中的部分。

5.2.2　调试并完善脚本

1．工作任务描述

对录制好的基础脚本，按照测试场景的需要进一步调试及完善。

2．工作过程

（1）运行/调试测试脚本

① 经过以上的各个步骤后，脚本就可以运行了。运行脚本可以通过菜单或者工具栏来操作。如图 5-17 所示。

② 执行“运行”命令后，VuGen 先编译脚本，检查是否有语法等错误。如果有错误，VuGen 将在“Execution Log”栏中显示错误信息。双击错误信息，VuGen 能够定位到出现错误的那一行脚本。为了验证脚本的正确性，我们还可以调试脚本，比如在脚本中加断点等，操作和在 VC 中完全一样，相信大家谁都不会感到陌生。

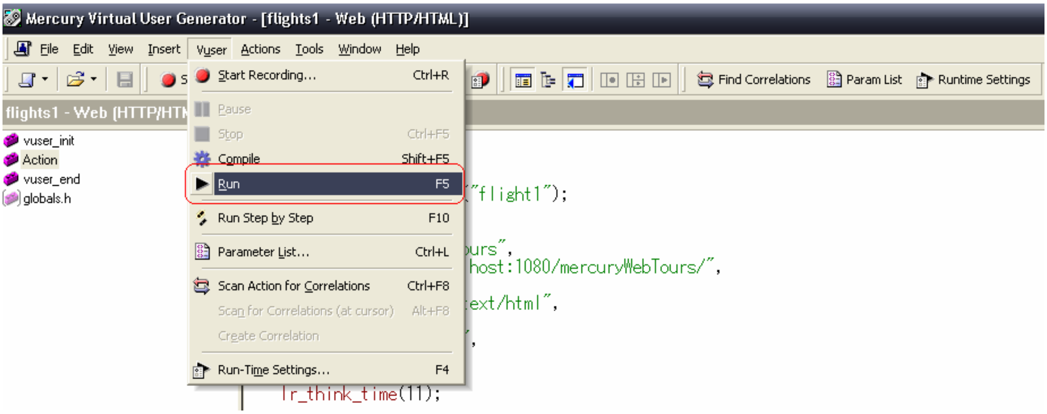


图 5-17 运行测试脚本菜单

（2）完善测试脚本

为什么要完善测试脚本呢？

首先，为了衡量服务器的性能，需要定义事务（Transaction）。例如在脚本中有一个数据查询操作，为了衡量服务器执行查询操作的性能，可以把这个操作定义为一个事务。这样在运行测试脚本时，LoadRunner 运行到该事务的开始点时，就会开始计时，直到运行到该事务的结束点，计时结束。这个事务的运行时间在测试结果中会有反映。LoadRunner 允许在脚本中插入不限数量的事务。

在方案执行期间，控制台将测量执行每个事务所用的时间。方案运行后，可使用 LoadRunner 的图和报告来分析各个事务的服务器性能。

其次，使用集合点是为了衡量在加重负载的情况下服务器的性能情况。在测试计划中，可能会要求系统能够承受多人同时提交数据，LoadRunner 通过在提交数据操作前面加入集合点的方法，检查同时有多少用户运行到集合点，人数不足时，LoadRunner 会命令已经到集合点的用户等待，当在集合点等待的用户达到要求容纳的人数（如 1000 人）时，LoadRunner 向系统提交数据。

在脚本中加入集合点后，控制台运行脚本时，可以对集合点进行策略设置，这样就可以根据实际情况在系统上模拟用户负载了。

再次，在录制过程中最好加入注释，因为在录制完脚本后看到的都是脚本代码，操作复杂的业务无法找到相应的位置进行关联或者参数化的动作，这时，注释就显得尤为重要。

最后，LoadRunner 提供了很多函数，有些函数是在录制时根据不同的协议自带的函数。其中有些函数是供手工添加的，这就要根据实际情况进行添加了。例如脚本关联，有些变量无法实现系统自动关联，只能添加函数进行手动关联。

在录制完成的脚本中，还可以根据实际情况，添加事务、集合点、注释、函数等内容来增强脚本，进一步完善。下面逐一进行介绍。

① 插入事务

事务：为了衡量服务器的性能，我们需要定义事务。比如：我们在脚本中有一个向服务器提交数据的操作，为了衡量服务器处理该操作的性能，我们把这个操作定义为一个事务，这样在运行测试脚本时，LoadRunner 运行到该事务的开始点时，LoadRunner 就会开始计时，直到运行到该事务的结束点，计时结束。这个事务的运行时间在结果中会有反映。插入事务

操作可以在录制过程中进行，也可以在录制结束后进行。LoadRunner 运行在脚本中插入不限数量的事务。

具体的操作方法如下：在需要定义事务的操作前面，通过菜单或者工具栏插入。如图 5-18 所示。

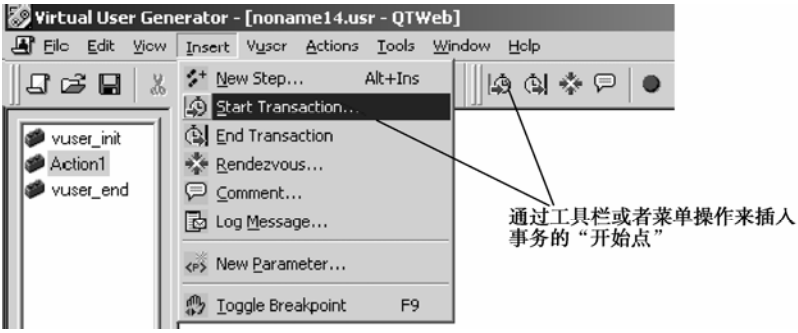


图 5-18 插入事务的开始点菜单

出现以下对话框，如图 5-19 所示。

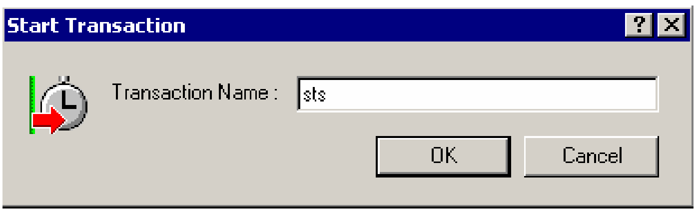


图 5-19 事务名称对话框

输入该事务的名称。注意：事务的名称最好要有意义，能够清楚地说明该事务完成的动作。插入事务的开始点后，下面需要在需要定义事务的操作后面插入事务的“结束点”。同样可以通过菜单或者工具栏插入。如图 5-20 所示。

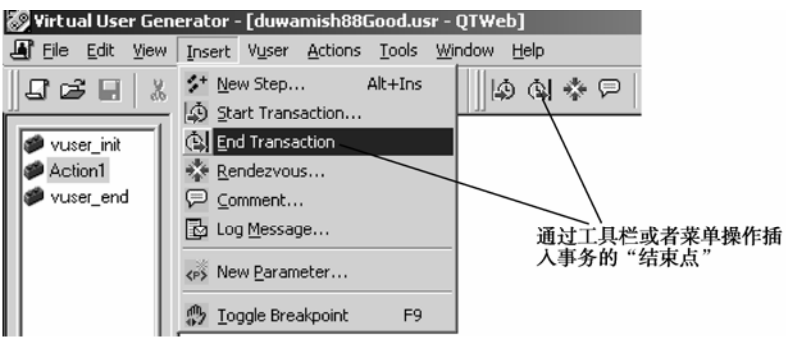


图 5-20 事务的结束点菜单

出现以下对话框，如图 5-21 所示。

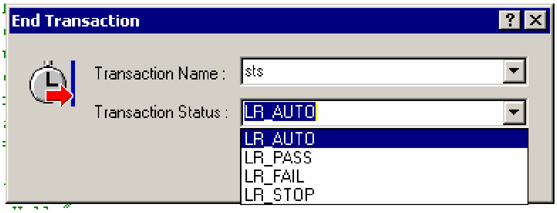


图 5-21　结束对话框

默认情况下，事务的名称列出最近的一个事务名称。一般情况下，事务名称不用修改。

事务的状态默认情况下是 LR_AUTO。一般情况下，我们也不需要修改，除非在手工编写代码时，有可能需要手动设置事务的状态。

脚本中事务的代码如下：

```
lr_start_transaction("sts");
/*
具体的事务
*/
......
      lr_end_transaction("sts", LR_AUTO);
```

② 插入集合点

插入集合点是为了衡量在加重负载的情况下服务器的性能情况。在测试计划中，可能会要求系统能够承受 500 人同时提交数据，在 LoadRunner 中可以通过在提交数据操作前面加入集合点，这样当虚拟用户运行到提交数据的集合点时，LoadRunner 就会检查同时有多少用户运行到集合点，如果不到 500 人，LoadRunner 就会命令已经到集合点的用户在此等待，当在集合点等待的用户达到 500 人时，LoadRunner 命令 500 人同时去提交数据，从而达到测试计划中的需求。

注意：集合点经常和事务结合起来使用。集合点只能插入到 Action 部分，vuser_init 和 vuser_end 中不能插入集合点。

具体的操作方法如下：在需要插入集合点的前面，通过菜单或者工具栏操作，如图 5-22 所示。



图 5-22　插入集合点窗口

出现对话框，如图 5-23 所示。

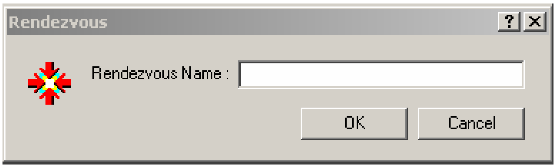


图 5-23　集合点名称对话框

输入该集合点的名称。注意：集合点的名称最好要有意义，能够清楚地说明该集合点完成的动作。

脚本中集合点的代码如下：

```
lr_rendezvous("sts_server");
```

③ 参数化输入

如果用户在录制脚本过程中，填写提交了一些数据，比如要增加数据库记录。这些操作都被记录到了脚本中。当多个虚拟用户运行脚本时，都会提交相同的记录，这样不符合实际的运行情况，而且有可能引起冲突。为了更加真实地模拟实际环境，需要各种各样的输入。

参数化输入是一种不错的方法。

用参数表示用户的脚本有两个优点：

a．脚本的长度变短。

b．使用不同的数值来测试你的应用。

参数化包含以下两项任务：

c．在脚本中用参数取代常量值。

d．设置参数的属性以及数据源。

参数化仅可以用于一个函数中的参量。你不能用参数表示非函数参数的字符串。另外，不是所有的函数都可以参数化的。

参数化输入的讲解，我们采用一个例子的方式来进行。

```
lr_start_transaction("sts");

web_set_certificate_ex("CertIndex=1",
    LAST);
lr_rendezvous("sts_server");

web_url("app",
    "URL=https://172.16.8.123/app/",
    "TargetFrame=",
    "Resource=0",
    "RecContentType=text/html",
    "Referer=",
    "Snapshot=t1.inf",
    "Mode=HTML",
    LAST);
web_find("web_find",
    "What=Hello",
    LAST);

lr_end_transaction("sts", LR_AUTO);
```

假如有以上的一个提交数据的窗体，我们想参数化高亮显示的部分（1），如图 5-24 所示。操作方法很简单，我们只要选中“1”，然后单击鼠标右键。

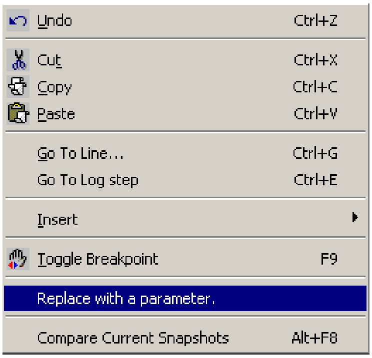


图 5-24　参数替换菜单

出现如下对话框，如图 5-25 所示。

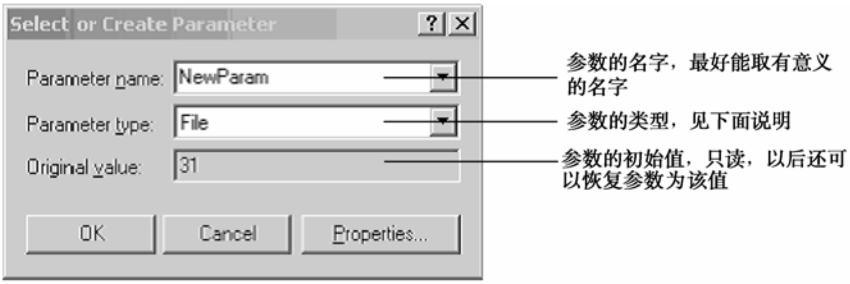


图 5-25　输入参数信息对话框

下面我们重点介绍一下参数的类型。

- DateTime：很简单，在需要输入日期/时间的地方，可以用 DateTime 类型来替代。其属性设置也很简单，选择一种格式即可。当然也可以定制格式。
- Load Generator Name：在实际运行中，LoadRunner 使用该虚拟用户所在 Load Generator 的机器名来代替。
- Iteration Number：在实际运行中，LoadRunner 使用该测试脚本当前循环的次数来代替。
- Random Number：随机数。很简单。在属性设置中可以设置产生随机数的范围
- Unique Number：唯一的数。在属性设置中可以设置第一个数以及递增的数的大小。

注意：使用该参数类型必须注意可以接受的最大数。例如：某个文本框能接受的最大数为 99。当使用该参数类型时，设置第一个数为 1，递增的数为 1，但 100 个虚拟用户同时运行时，第 100 个虚拟用户输入的将是 100，这样脚本运行将会出错。注意：这里说的递增意思是各个用户取第一个值的递增数，每个用户相邻的两次循环之间的差值为 1。举例说明：假如起始数为 1，递增为 5，那么第一个用户第一次循环取值 1，第二次循环取值 2；第二个用户第一次循环取值为 6，第二次为 7；依次类推。

- Vuser ID：设置比较简单。在实际运行中，LoadRunner 使用该虚拟用户的 ID 来代替，该 ID 是由 Controller 来控制的。
- File：需要在属性设置中编辑文件，添加内容，也可以从现成的数据库中取数据。
- User Defined Function：从用户开发的 dll 文件提取数据。

上面的例子中，我们取随机数即可。如图 5-26 所示。

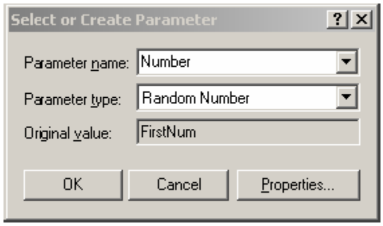


图 5-26　参数信息窗口

单击［Properties…］按钮，进入属性设置对话框，如图 5-27 所示。

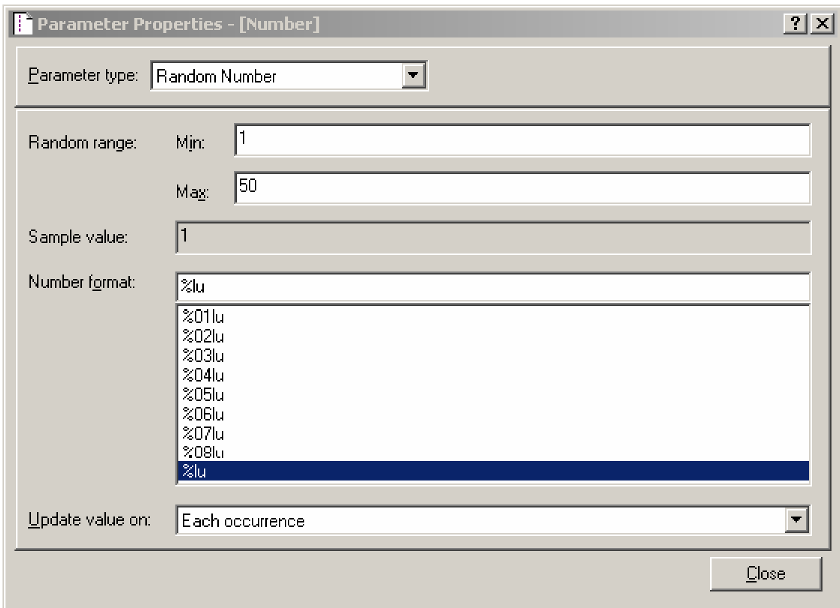


图 5-27　属性设置对话框

添入随机数的取值范围为（1～50），选择一种数据格式。在 Update Value on 中有以下几个选项：

- Each Occurrence：在运行时，每遇到一次该参数，便会取一个新的值
- Each iteration：运行时，在每一次循环中都取相同的值
- Once：运行时，在每次循环中，该参数只取一次值

④ 插入检查点：

在进行压力测试时，为了检查 Web 服务器返回的网页是否正确，VuGen 允许我们插入 Text/Imag 检查点，这些检查点验证网页上是否存在指定的 Text 或者 Imag，还可以测试在比较大的压力测试环境中，被测的网站功能是否保持正确。插入检查点时，将代码显示方式切换成 TreeView 视图是比较方便的，插入检查点可以在脚本录制过程中进行，也可以在录制完成后进行，推荐最好能在录制过程中添加 Text/Imag 检查点。具体操作如下：

a．将代码显示方式切换到 TreeView 视图，如图 5-28 所示，首先在树形菜单中选择需要插入检查点的一项，单击鼠标右键，选择将检查点插入到该操作执行前还是该操作执行后。如果在该操作执行前，则选择“Insert Before”，否则选择“Insert After”。

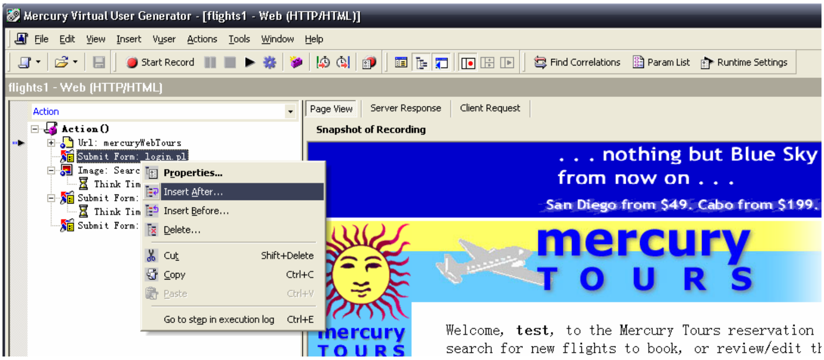


图 5-28　插入检查点

b. 例如选择“Insert After”，将弹出如图 5-29 所示的对话框，在对话框中选择“Text Check”（这里以 Text 检查点为例说明），单击［OK］按钮，将弹出“Text Check Properties”对话框。

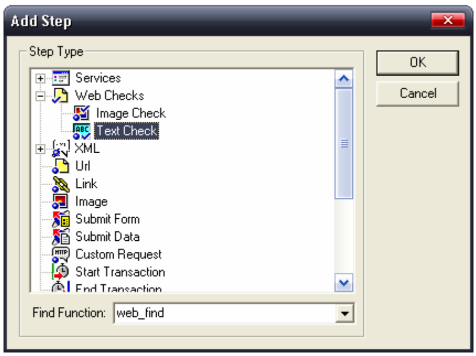


图 5-29　选择 Text 检查点对话框

c. 如图 5-30 所示，在“Text Check Properties”对话框的“Search for”文本框中输入在实际操作界面中要搜索的字符串（如：Denver），如果要搜索的字符串在操作界面中可能存在多处，那么为了强调只搜索在某个另外字符串（如：to）右侧的“Denver”，则需要在“Right of:”文本框中输入标记性字符串“to”，左侧（Left of:）同上。

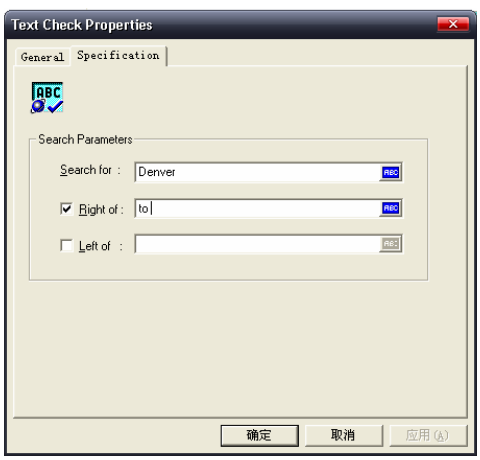


图 5-30　搜索字符串对话框

切换到“General”TAB 页，如图 5-31 所示，输入该操作步骤名称，单击［确定］按钮，即可完成添加 Text 检查点的任务。

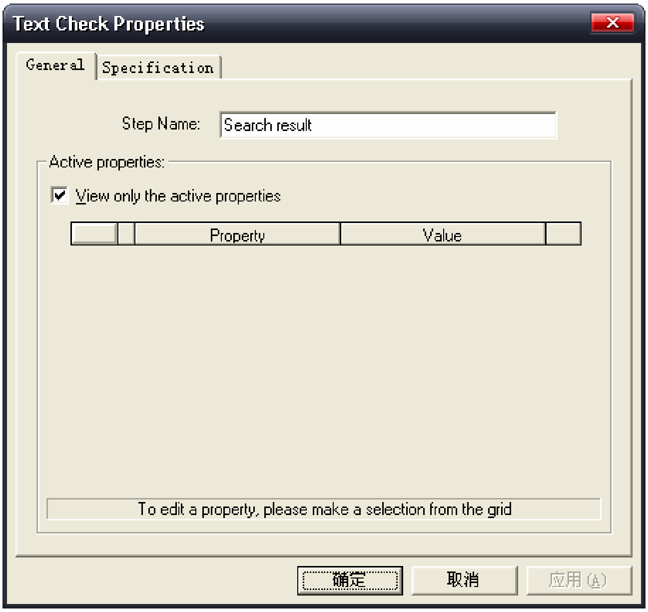


图 5-31　完成添加 Text 检查点对话框

5.2.3　应知应会

在录制脚本时要遵循以下录制原则：

（1）提高脚本执行效率

所录制的脚本内容要精练，而且是用户的真实操作，不可增加多余或重复性的操作，这样的脚本执行起来更能准确地模拟用户的真实行为，减少了执行时间，执行结果更准确。

（2）录制具有代表性的功能

在一个软件中有很多不同的功能，但要录制所有的功能几乎是不可能的，所以要选择常用的、使用频率较高的业务功能来进行测试。

（3）选择具有影响的事务

测试人员要对被测功能具有一定的认识和了解，选择一些对于整个测试过程中有影响的事务来测试，否则测试结果是无意义的。

5.2.4　拓展任务

对天天超市管理系统进行脚本的录制，并对脚本进行调试完善。

工作任务 5.3　软件测试场景

5.3.1　创建运行场景

1．工作任务描述

运行场景描述在测试活动中发生的各种事件。一个运行场景包括一个运行虚拟用户活动的 Load Generator 机器列表，一个测试脚本的列表以及大量的虚拟用户和虚拟用户组。创建运行场景使用 Controller。

在开始菜单中，启动 Controller 程序，出现“New Scenario”窗口。如果没有出现，可以在菜单或者工具栏中单击“New”。

在新建场景的窗口，选择一种场景类型。下面对三种类型进行简单的说明。

● Manual Scenario：该项要完全手动的设置场景。

● Manual Scenario with Percentage Mode：该项只有在“Manual Scenario”选中的情况下才能选择。选择该项后，在场景中我们需要定义要使用的虚拟用户的总数，Load Generator machine 机器集，然后我们为每一个脚本分配要运行的虚拟用户的百分比。

● Goal-Oriented Scenario：在测试计划中，一般都包括性能测试要达到的目标。选择该项后，LoadRunner 基于这个目标，自动为你创建一个场景。在场景中，我们只要定义好我们的目标即可。

2．工作过程

（1）选择场景类型为 Manual Scenario

① 选择 Vuser Groups，进入新建场景的界面，如图 5-32 所示。

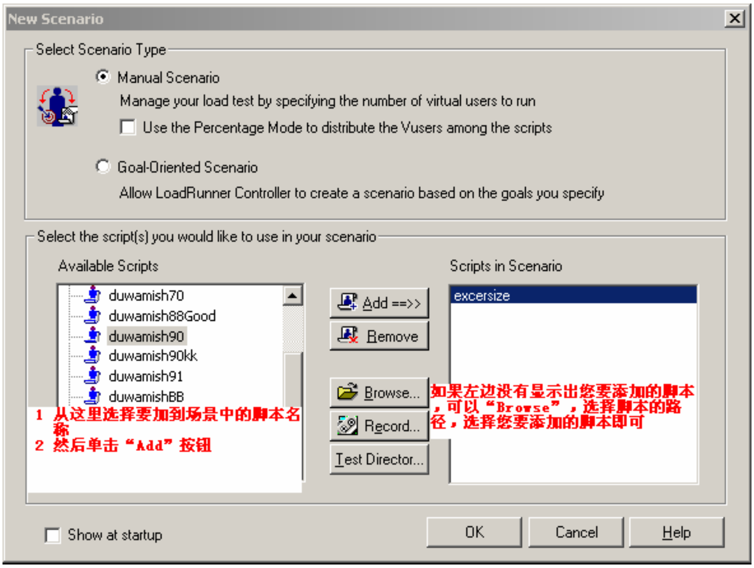


图 5-32　新建场景的界面

② 添加 Load Generator Machines

单击右边的［Generators］按钮，出现如图 5-33 的界面。

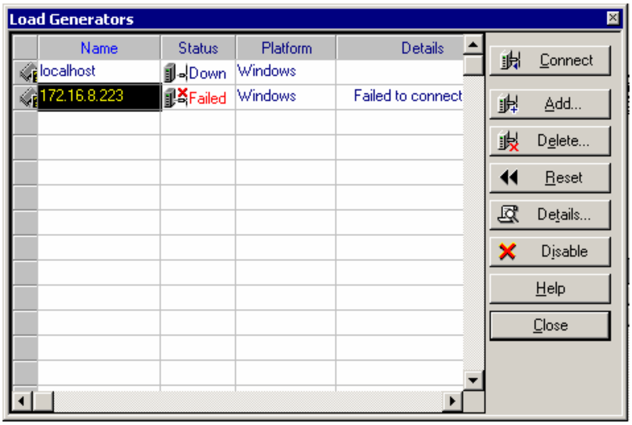


图 5-33　压力发生器界面

③ 添加 LoadGenerator 后，执行“Connect”操作，使 Status 为 Ready，表示该机器联接正常，如果为 Failed，表示该机器不能联接，请检查原因。可以把这个列表保存下来，如图 5-34 所示，执行菜单命令即可。

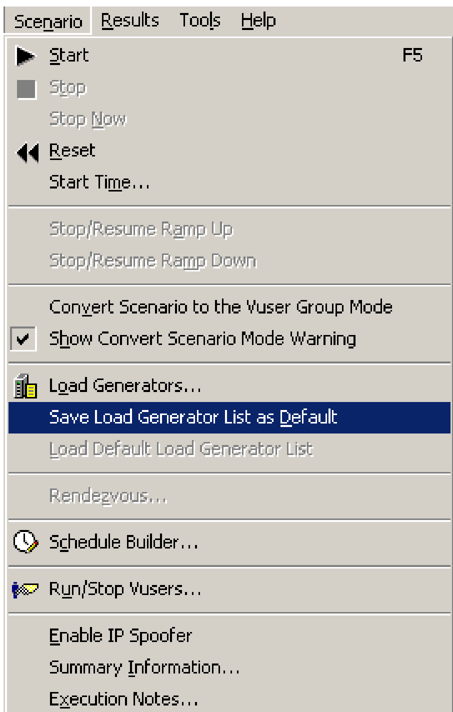


图 5-34　保存列表菜单

④ 添加虚拟用户

首先设置虚拟用户总数。虚拟用户总数界面，如图 5-35 所示。



图 5-35　设置虚拟用户总数界面

⑤ 设置 Schedule

这里的设置是非常重要的，也是三种场景类型最重要的区别之处。单击 Edit Schedule... 按钮，即可进入 Schedule 设置的界面，分别如图 5-36、图 5-37、图 5-38 所示。

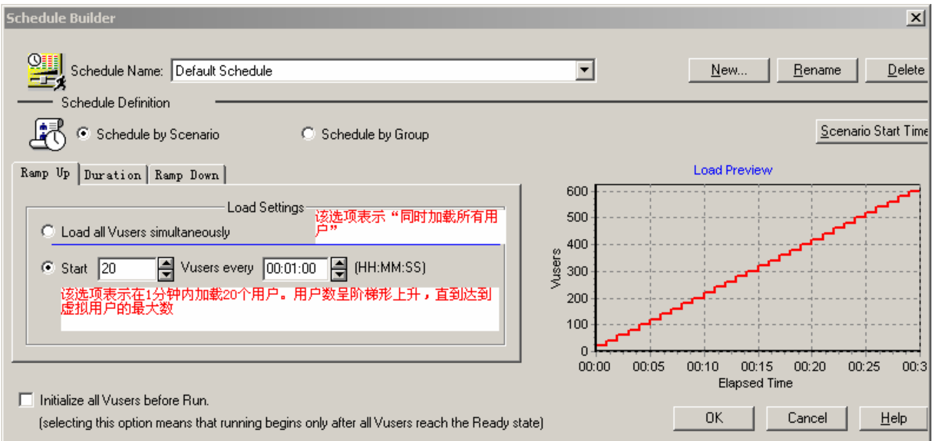


图 5-36　Ramp Up 界面

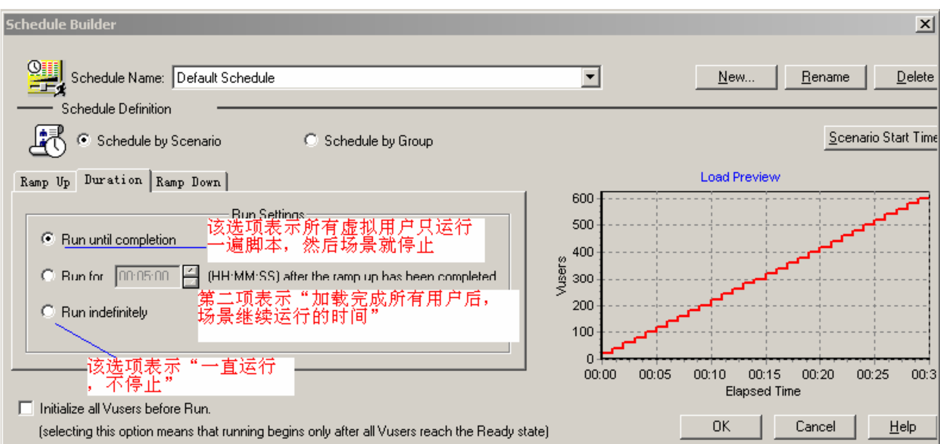


图 5-37　Duration 界面

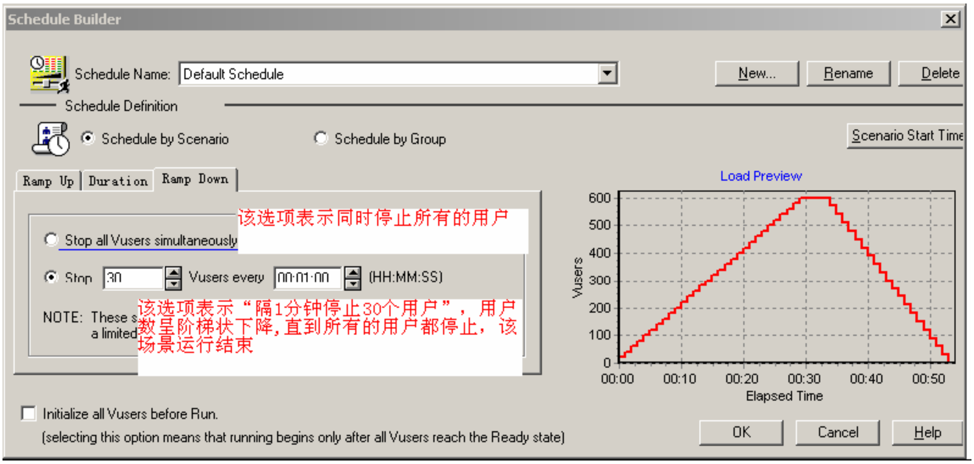


图 5-38 Ramp Down 界面

⑥ 单击 Scenario Start Time 按钮，进入 Scenario Start Time 界面，如图 5-39 所示。

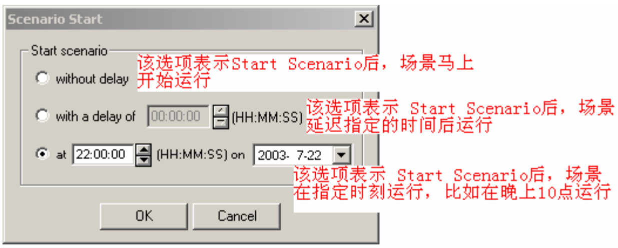


图 5-39 场景运行参数设置界面

⑦ 设置集合点

如果在脚本中设置了集合点，还需要在 Controller 中设置集合点策略。

在菜单中调出设置集合点策略的界面，如图 5-40 所示。单击 Rendezvows 命令，出现集合点状态界面，如图 5-41 所示。

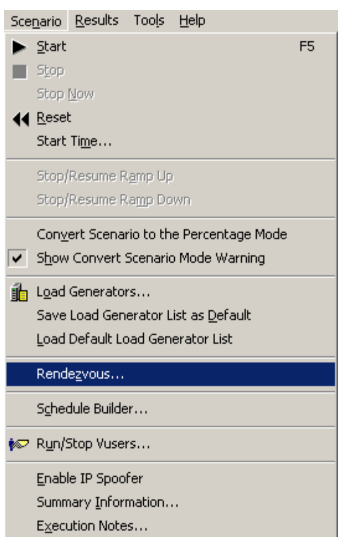


图 5-40 设置集合点策略菜单

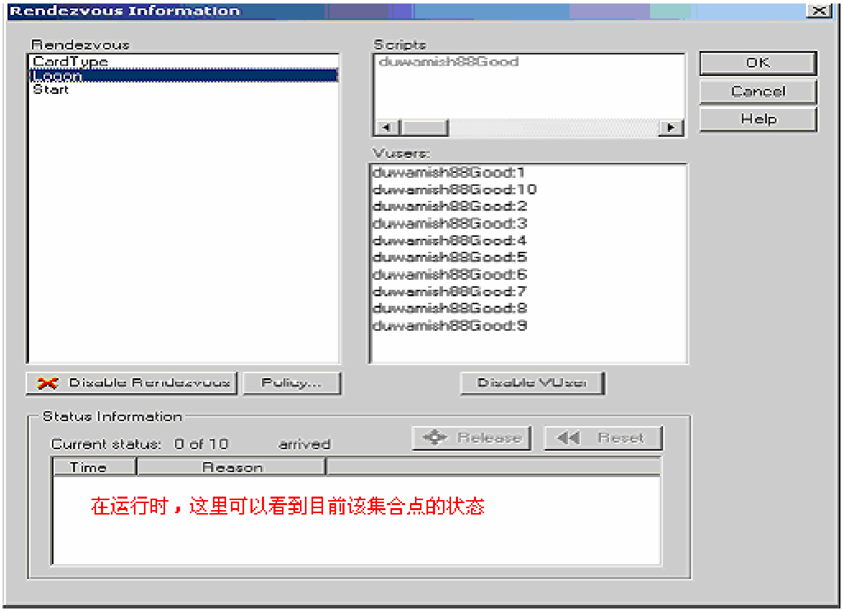


图 5-41　集合点状态界面

⑧ 单击 Policy... 按钮，进入策略设置界面，如图 5-42 所示。

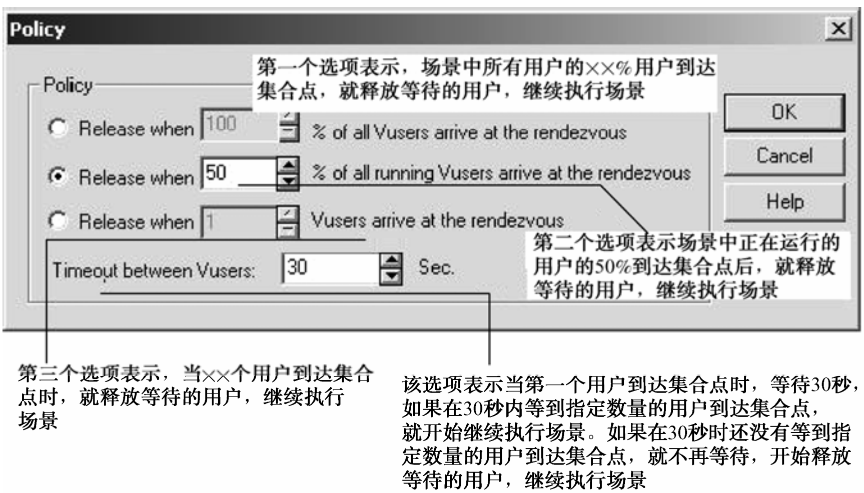


图 5-42　设置集合点策略界面

⑨ 设置结果文件保存路径，通过菜单操作，如图 5-43 所示。

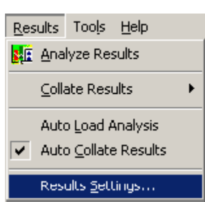


图 5-43　结果文件保存菜单

⑩ 调出结果文件的保存路径，如图 5-44 所示。

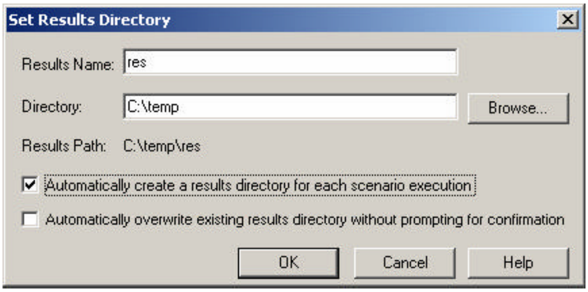


图 5-44　保存路径界面

⑪ Run-Time Setting

使用同上。

（2）Manual Scenario with Percentage Mode

该场景类型和“Manual Scenario”类型非常类似，下面简单地对它们不一样的地方进行设置。场景编辑界面，如图 5-45 所示，场景描述界面，如图 5-46 所示。

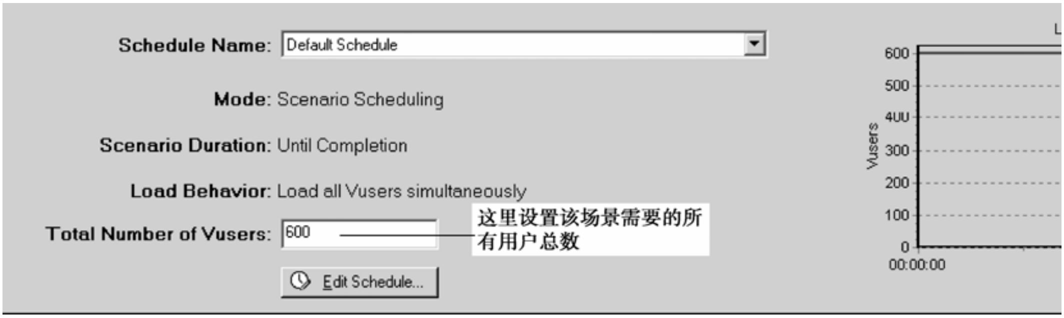


图 5-45　场景编辑界面



图 5-46　场景描述界面

（3）选择场景类型为 Goal—Oriented Scenario

单击按钮，编辑该场景的目标界面，如图 5-47 所示，加载用户界面，如图 5-48 所示。

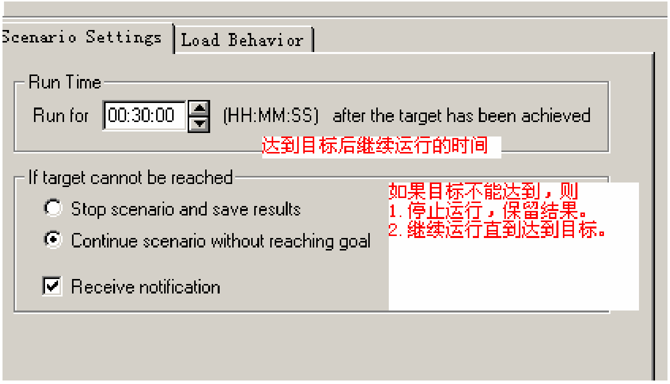


图 5-47　编辑场景目标界面

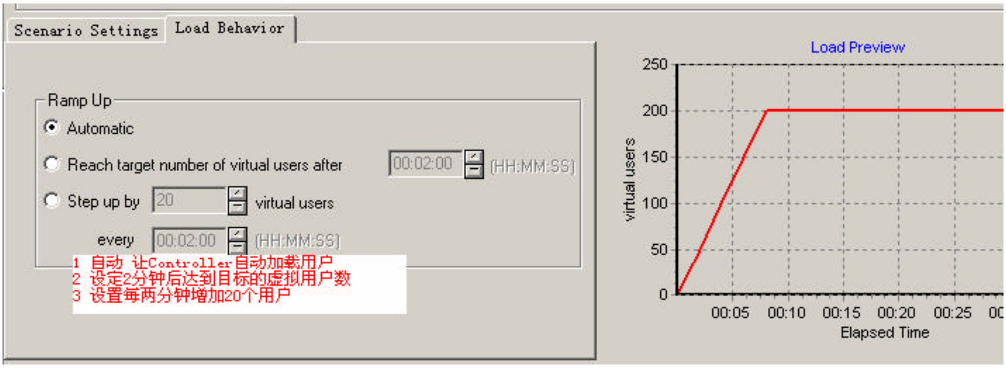


图 5-48　加载用户界面

注：以上的说明是以选择的目标为 Virtual Users 时为基础的。选择不同的目标，内容会稍微有一点不同。

（4）目标的种类。每次场景运行只能设置一个目标，目标类型界面，如图 5-49 所示。

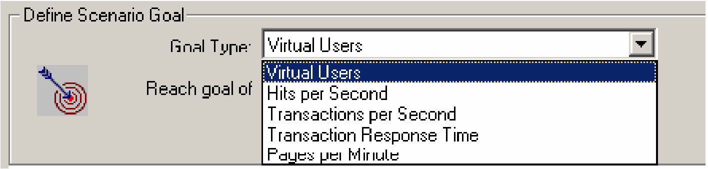


图 5-49　目标类型界面

① Virtual Users Goal

如果需要测试多少人可以同时运行 Web 应用，那么推荐定义 Virtual Users Goal。运行定义该目标类型的场景，如图 5-50 所示，和运行 Manual 类型的场景类似。

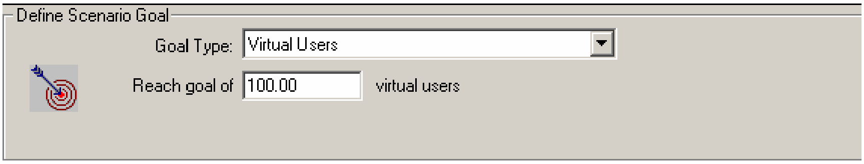


图 5-50　Virtual Users 类型界面

② Hits per Second

如果想测试 Web Server 的真正实力，推荐定义目标类型为：Hits per Second、Pages per Minute 或者 Transactions per Second，这些类型都需要指定一个虚拟用户的最小值和最大值的范围。

Controller 试图使用最少的虚拟用户来达到定义的目标。如果使用最少的用户，不能达到目标，Controller 增加用户数，直到定义的最大值。如果使用了最多的虚拟用户数，定义的目标还没有实现，那么需要增加最大用户数，重新执行场景，设置虚拟用户界面如图 5-51 所示。

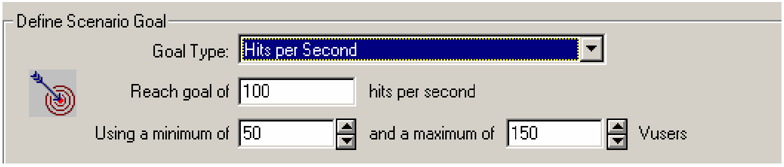


图 5-51　设置虚拟用户界面

③ Transactions per Second

在 Transaction Name 下拉列表中选择事务，设置事务名称界面如图 5-52 所示。

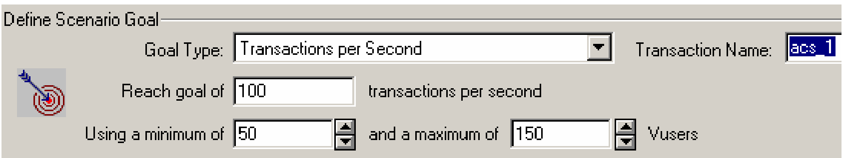


图 5-52　设置事务名称界面

④ Transactions Response Time

如果想知道在多少用户并发访问网站时，事务的响应时间达到性能指标说明书中规定响应时间的最大值，那么推荐使用 Transactions Response Time 类型。指定需要测试的事务的名称，虚拟用户数量的最小值和最大值，还有预先定义好的事务的响应时间。

在场景运行中，如果使用了最多的虚拟用户，还不能达到定义的最大响应时间，说明 Web Server 还有能力接纳定义的虚拟用户的最多数量，设置虚拟用户界面如图 5-53 所示。如果在使用了部分虚拟用户，就达到了定义的最大的响应时间，或者 LoadRunner 提示如果使用最多数量的虚拟用户时将要超过最大响应时间，那么需要重新设计或者修补应用程序，同时可能需要升级 Web Server 的软硬件。事务响应时间界面如图 5-54 所示。

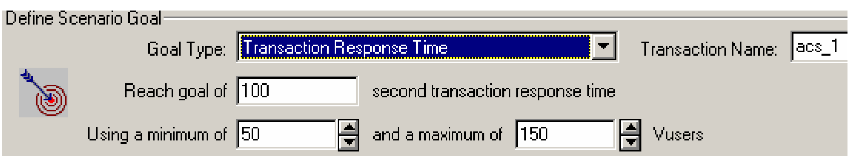


图 5-53　设置虚拟用户界面

⑤ Pages per Minute

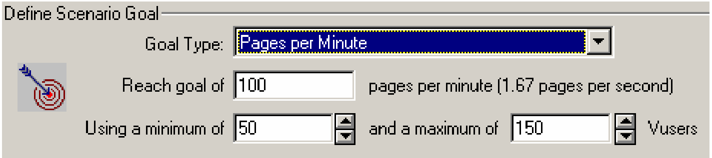


图 5-54 事务响应时间界面

⑥ 理解各种类型

如果你定义的类型是 Pages per Minute、Hits/Transactions per Second，Controller 首先用最小用户数除以定义的目标，得到一个值，然后确定每个用户应该达到的 hits/transactions 或者 pages per minute，然后 controller 开始按照以下的策略加载用户。

如果选择的是自动的加载虚拟用户，LoadRunner 会首先加载 50 个用户。如果定义的最大用户数小于 50，LoadRunner 就会一次加载所有的虚拟用户。

a．如果选择的是在场景运行一段时间后达到目标，LoadRunner 就会尝试在定义的这段时间内达到目标，根据时间限制和计算出的每个用户的 hits、transactions 或者 pages，LoadRunner 确定第一批加载多少用户。

b．如果选择的是按照一定的阶段达到目标（也就是先在 x 长时间内达到 y pages/hits，然后再达到下一个目标），LoadRunner 计算每个用户应该达到的数字后，再确定第一批加载多少用户。

每加载一批用户后，LoadRunner 会判断是否达到这批用户的目标。如果这批用户的目标没有达到，LoadRunner 重新计算每一个用户应该达到的目标数字后，重新调整下一批加载用户的数量。默认情况下，LoadRunner 每两分钟加载一批用户。

如果 Controller 加载了最多数量的用户还没有达到预定的目标，LoadRunner 会重新计算每个用户的目标，然后同时运行最大数量的用户，尝试达到预定的目标。

如果出现以下情况，Pages per Minute、Hits/Transactions per Second 类型的场景会置于“Failed”状态。

c．Controller 使用了指定的最大数量的用户，并且两次都没有达到目标。

d．所有的用户运行都失败。

e．没有足够的 Load Generators 机器（现有的机器已经超载运行的情况下）。

f．Controller 增加了几批用户后，pages per minute 或者 hits/transactions per second 没有增加。

g．Controllller 加载第一批用户后，定义的目标没有被捕捉到。

5.3.2 应知应会

启用 IP Spoofer（IP 欺骗）

当运行场景时，虚拟用户使用它们所在的 Load Generator 的固定的 IP 地址。同时每个 Load Generator 上运行大量的虚拟用户，这样就造成了大量的用户使用同一 IP 同时访问一个网站的情况，这种情况和实际运行的情况不符，并且有一些网站会根据用户 IP 来分配资源，这些网站会限制同一个 IP 的登录，使用等。为了更加真实的模拟实际情况，LoadRunner 允许运行的虚拟用户使用不同的 IP 访问统一网站，这种技术称为“IP 欺骗”。

启用该选项后，场景中运行的虚拟用户将模拟从不同的 IP 地址发送请求。该选项非常的

有用。注意：IP Spoofer 在连接 Load Generators 之前启用。

要使用 IP 欺骗，各个 Load Generator 机器必须使用固定的 IP，不能使用动态 IP（即 DHCP）。

使用 IP Spoofer 的步骤如下：

（1）使用 IP Wizard：在“开始”菜单程序中，找到 LoadRunner——Tools——IP Wizard，IP Wizard 运行界面如图 5-55 所示。运行它注意：运行 IP Wizard 程序的机器必须使用固定的 IP，不能使用动态 IP。

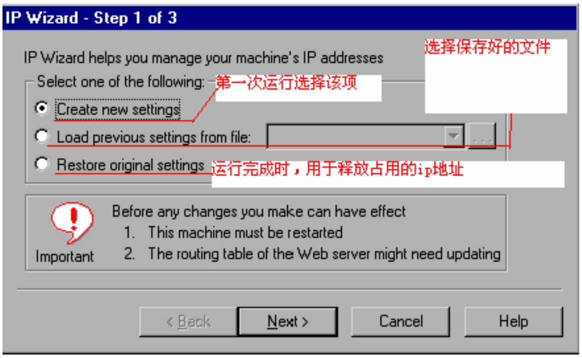


图 5-55　IP Wizard 运行界面

这里选择第一项，单击［Next］按钮，出现 IP Wizard 的输入 IP 界面，如图 5-56 所示。

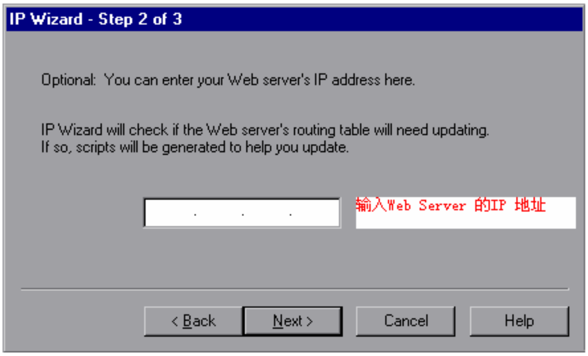


图 5-56　IP Wizard 输入 IP 界面

然后单击［Next］按钮，出现向导的第三个界面。

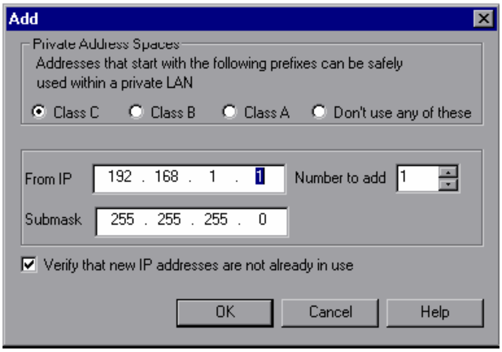


图 5-57　添加信息界面

从“From IP”文本框中输入要使用 IP 范围的第一个 IP 值，然后在“Numbers to Add”输入一个数字，表示 IP 范围的值；假如第一个 IP 为 192.168.6.100，范围大小为 100，那么 IP Wizard 将会使用 192.168.6.N（100<=N<200），当然这个范围内已经使用的 IP 地址除外，否则会引起 IP 冲突。“Submask”采用默认情况即可，取决于使用的那种类型的网络 IP，一般局域网内采用 Class C 即可。然后“OK”，然后 IP Wizard 开始检查该范围内没有使用的 IP，并把没有使用的 IP 添加到本机的 IP 窗口中。到最后一个窗口，直接点“Finish”，使用 IP Wizard 后，最后重新启动机器。

（2）在 Controller 的场景中，启用 IP Spoofer 即可，如图 5-58 所示。

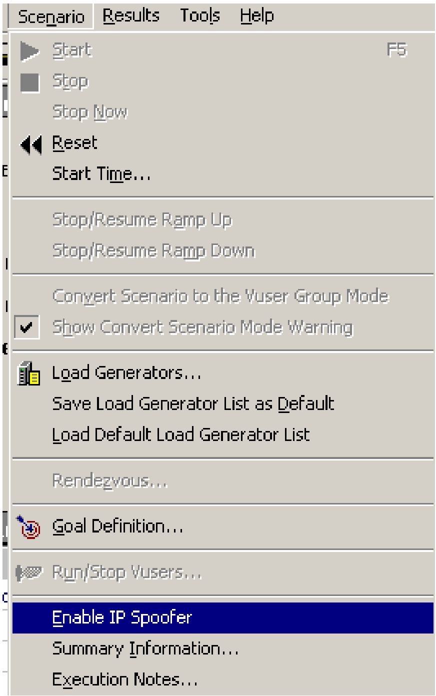


图 5-58　IP Spoofer 菜单

5.3.3　拓展任务

独立设置天天超市的测试场景，独立进行性能测试活动。

工作任务 5.4　LoadRunner的结果分析

5.4.1　LoadRunner调用Analysis

1．工作任务描述

场景运行结束后，可以使用 Analysis 组件分析结果，以下为具体操作过程。注意：这里介绍的分析方法只适用于 Web 测试。

2．工作过程

（1）利用 Controller，按照设定的测试场景运行测试脚本后，如图 5-59 所示，在“Mercury

LoadRunner Controller”窗口中，单击菜单“Results”选择“Analyze Results”菜单项，将启动“Mercury LoadRunner Analyze”对当前测试结果进行分析。

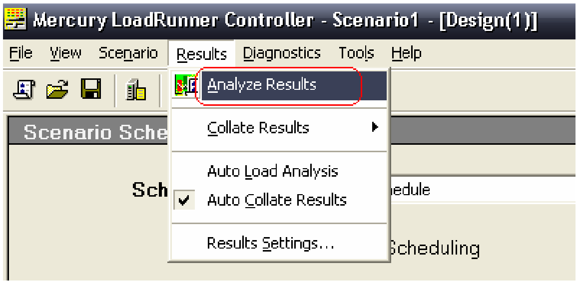


图 5-59　场景设置窗口

（2）在“Mercury LoadRunner Analyze”中，将自动统计出当前测试结果的各项数据，例如：Transaction Summary、Average Transaction Response Time 等，如图 5-60 所示，在“Mercury LoadRunner Analyze”主窗口，单击左侧视图中的目录项，在右侧数据视图中，将以表格、曲线图以及柱状图等方式显示对应的测试结果。

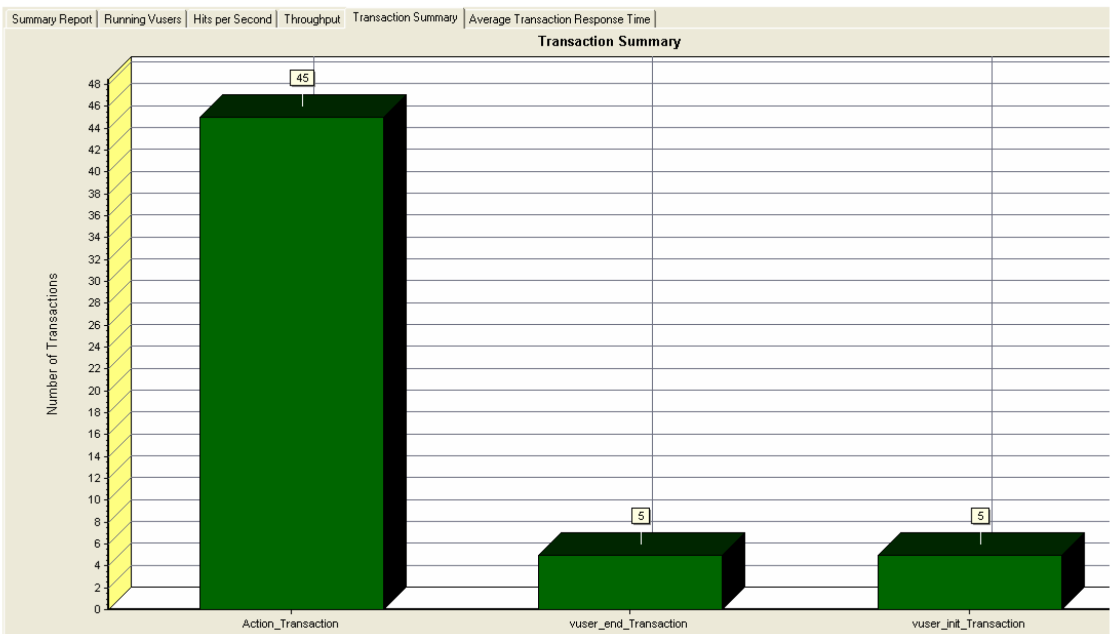


图 5-60　完成事务合计报表界面

（3）如果默认显示的图表数据结果，不能满足需要，还可以增加新的表格来显示所需要的报告项目，操作如下：

① 单击“Mercury LoadRunner Analyze”主窗口的“Graph”菜单，选择“Add Graph”，如图 5-61 所示，将弹出“Open a New Graph”对话框。

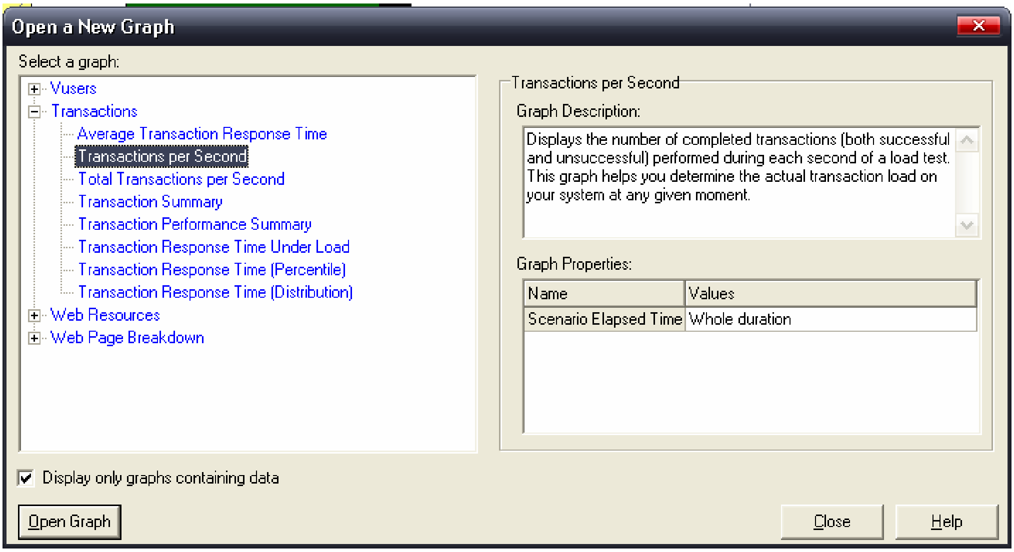


图 5-61　打开新图表对话框

② 在“Open a New Graph”对话框左侧树状目录中，选择要显示的图表项目，单击［Open Graph］按钮，新增显示图表完成。

（4）生成测试报告

完成上一步结果分析，便可利用“Mercury LoadRunner Analyze”以多种格式生成测试报告，操作如下：

在“Mercury LoadRunner Analyze”主窗口，单击“Reports”菜单，选择“HTML Report”菜单项，将生成 HTML 格式的测试报告，选择“Microsoft Word Report”菜单项，将生成 Word 格式的测试报告。

5.4.2　应知应会

1．测试报告的基本要素

（1）Bug 说明：（what）

（2）Bug 出现的地方：（where）

（3）分析 Bug：（why）

（4）解决方法：（how）

（5）报告人：（who）

2．软件测试报告模板

×××公司

×××（产品或软件）/×××（模块）测试报告

（1）概述

测试目的　简述本次测试的目的，如：验证某模块是否符合设计

项目背景　简述测试所在项目的背景，如：×××（项目）目前进入什么阶段，以及其他信息

（2）测试环境

硬件环境　仅针对测试对象的硬件环境及其版本信息加以说明

软件环境　仅针对测试对象的软件环境及其版本信息加以说明

（3）测试人员

人员

角色

（4）实际进度

占用时间　描述整个测试过程的时间跨度，如：xxxx-xx-xx 至 xxxx-xx-xx

进度情况　原因　如果测试提前或延后完成，请说明具体原因

（5）测试参考文档

《×××测试计划》

《×××测试用例》

《文档三》

《文档四》

版本信息 V1.0

（6）测试数据

测试数据

测试项总数 0

PASS 0 PASS 率　#DIV/0!

FAIL 0 FAIL 率　#DIV/0!

严重度——高　0　其中：高—#DIV/0!

严重度——中　0　中—#DIV/0!

严重度——低　0　低—#DIV/0!

测试项编号　测试项　通过与否　问题描述　问题严重度

注：问题严重度的界定：

高——导致系统死机或后续部分测试项功能不能实现；

中——影响该部分的测试功能的完整性且急需解决；

低——仅属于系统中的小 Bug，或根据测试过程发现的需要调整的部分，但并非急需解决。

（7）项目的总结　对整个测试项目进行总结性阐述，如：测试是否通过，导致 FAIL 的主要原因。

（8）意见和建议　针对本次测试工作，提出自己的意见或建议。没有可填“无”。

5.4.3　拓展任务

运用 LoadRunner 对天天超市管理系统测试结果进行分析。

附录A　常用测试工具清单

数据/测试性能类：

LoadRunner

LoadRunner 是一款优秀的压力和性能测试工具，可以模拟成千上万的并发操作，对应用系统、Web Service、Web 服务器、数据库等进行压力和性能测试，兼容 Windows 和 UNIX。

File-AID

File-AID 是一个企业级的数据管理工具，测试人员可以利用它来快速地构建测试数据环境，支持 mainframes、MVS、DB2 和分布系统。

SQL Data Generator

Red Gate 的 SQL Data Generator 替代 Intel 的 Vtune Performance Analyzer 的位置，是一颗璀璨的新星，宣称可以在一杯咖啡的时间内，为 10 个表格创建 2 百万行数据。

功能测试类：

1．QTP

QTP 是 Windows 平台下出色的自动化功能测试和回归测试工具，基于 GUI 的录制和回放测试，加上 VBScript，测试人员可以轻易控制和操纵程序界面对象，创建自动化测试用例。

2．Rational Functional Tester

Rational Functional Tester 的特点是，除了自身的脚本开发环境外，还支持两种开发环境：Eclipse 框架中的 Java、Microsoft Visual Studio 中的 Visual Basic.NET。

3．SilkTest

SilkTest 同样是一款不错的自动化功能测试和回归测试工具，支持 C/S 结构的 Java、.NET 和 Web。

静态/动态代码分析类：

1．Rational Software Analyzer Developer Edition

能够捕捉内存泄露、分析应用程序性能、代码覆盖率等，支持广泛的编程语言，包括 C/C++、Java、.NET、VB、VC++，支持 Linux、UNIX 和 Windows 平台。

2．TPTP

Eclipse Test and Performance Tools Platform（TPTP）替代了 Compuware 的 DevPartner Studio 的位置。Eclipse 的 TPTP 在新版本中添加了不少新特性。

3．DevInspect

HP 的 DevInspect 替代了 Parasoft 的 Jtest 的位置，2007 年年底收购了 SPI Dynamics，DevInspect 在自动化安全测试方面拥有很强的优势。

测试/QA 管理类：

1．SilkCentral Test Manager

用于简化跨平台的测试，还可管理 Junit/Nunit 等第三方的测试框架。

2．Optim Test Data Management Solution

作为 IBM 的测试数据管理和应用程序质量改进方案，Optim Test Data Management Solution 允许测试人员指定覆盖率标准、创建错误和边界条件、模拟产品环境等，支持 PeopleSoft 和 Siebel 等企业应用，支持 IBM、 Microsoft、Oracle、Sybase 等厂家的数据库，支持 Linux、UNIX、Windows 等平台。

3．TestDirector for Quality Center

包括需求管理、测试计划、测试执行和缺陷管理模块，全面管理了测试过程，成为测试人员最喜欢的测试管理工具。

4．TestDirector

基于 Web 的管理模式，允许测试人员和项目经理收集需求、设计和安排手工与自动化的测试、分析测试结果、生成图文并茂的测试报告，并且能与 HP 的其他功能测试工具 WinRunner、QTP 紧密结合。

缺陷/问题管理类：

Visual Studio Team Edition for Software Testers

该工具与 Team Fundation 结合起来，可以做到缺陷/问题跟踪自动化。这个工具还可以对 Web 应用程序和 Web 站点进行功能和压力测试。

附录B　基于测试概念进行代码设计的基本原则

当设计大型程序的时候，你必须时刻留心不同设计选项对诸如性能和可扩展性这样的特征的影响。随着软件产品的日渐复杂及其无所不在的部署，软件的“可测试性”也成了更重要的考虑事项。彻底测试代码的重要性是显然的。花在编写测试和测试代码上的时间和精力给你带来的回报是维护成本的大幅降低。然而，除非你很小心，否则你花在测试代码上的精力可能会首先达到花在编写代码上的精力的几倍！很多时候程序员们齐心协力地对他们的全部代码进行单元测试，结果花在上面的时间使大多数人都以沮丧而告终。

幸运的是，没有必要这样。在你设计软件的时候应用一些基本原则，编写易于测试、甚至使测试成为乐趣的代码是可能的。跟其他编码原则一样，这些原则也不是不容置疑或不可改变的教条。有时候打破这些规则也是必要的。因此，理解每条原则背后的动机和判断何时这些动机不适用（或应让位给更关心的问题）的能力是很重要的。

原则 1：到GUI（图形用户界面）视图的外面去

尽可能把代码移到GUI视图的外面。然后各种GUI动作就能成了模型上的简单方法调用。为什么你需要这样做呢？对GUI测试者来说，通过方法调用测试功能比间接地测试功能容易得多。另一个好处是它使修改程序功能而不影响视图变得更容易。当然，视图中也可能存在错误。在理想情况下，对程序的测试将同时检查模型和视图。

原则 2：使用类型进行错误检查

类型是你的朋友，要尽可能多地用类型系统自动检查错误。类型能在程序运行之前自动捕捉程序中的错误。没有静态类型检查的话，类型错误将作为破坏者逗留在你的程序中，直到恰当的执行路径碰巧把它揭露出来为止。最大限度地发挥使用类型的长处是棘手的。通常，一组数据结构可以在一个抽象级别上一起使用，或者被分出，成为一个单一的、更高抽象级别的一个新的相关数据类型。事实上，编程语言自身的历史可以看成是可以编程的抽象级别的逐渐提高。汇编语言提供了比特到整数和浮点数的抽象。接下来是记录和函数抽象，然后又是诸如对象、类、线程以及异常这样的抽象。在每一抽象级别上，达到与更高级别抽象一致的功能是可能的，但那实质上仅仅是耗费更多精力，冒更多的错误风险。在面向对象语言（其他现代语言也一样）中，一个程序员在设计抽象上有很大的灵活性。在哪个抽象级别上设计程序就成了基于折中的决定，比如由抽象级别提供的更多的健壮性和由于不能在更低抽象级别上工作而带来的表达性（有时是性能）的损失。通常，高级别抽象带来的健壮性和简单性的价值很少被其他考虑事项超过。

原则 3：使用调节器避免“故障线路”（fault line）

“故障线路”是指独立组件之间的接口，独立组件之间和组件与其相应子组件之间相比，很少有交互。这种故障线路的一个典型示例是GUI视图和它的模型之间的接口。其他示例包括在编译器中处理的不同阶段之间的接口或操作系统的内核和用户界面之间的接口。找出程序的故障线路，然后用具有转发功能的调节器快速访问聚合组件。沿着故障线路隔离测试每个组件通常更容易。但如果每个组件暴露的对象有很多，或者组件中你想测试的一些对象只

有通过多个嵌套引用才能访问，那么测试就会变得很乏味。不用隔离测试，而是拥有你在它上面调用你想测试的各种方法的单个调节器对象通常是有帮助的。这个对象然后能把这些方法调用转发到适当的地方。沿着相同线路，设计和自己的测试代码串联在一起的程序组件接口是有益的。这将使你把注意力集中在使这些接口尽可能简单上。

原则 4：使用小型签名和默认参数

使用小型方法说明和重载带默认方法参数的方法将使你在测试中调用这些方法变得愉快得多。否则，在测试这些方法时你将不得不构造额外参数。如果参数很大，那么将很快导致代码膨胀。更糟的是，它会使你编写比在其他情况下更少的测试。

原则 5：访问器不应修改内存状态

在测试中应该使用不修改内存状态的访问器来检查对象状态。在某些方面，测试和实验室试验相似，它们都想证明特定假设是有效的。如果特定检查动作改变了该领域的状态，那么要这样做会变得困难得多。与量子力学领域不同，计算机进程的状态可以不修改就被检查。使用这种原则是有好处的。

原则 6：用接口说明外部程序组件

用接口说明外部程序组件使得我们可以容易地在测试案例中模拟这些组件。这条原则能节省大量时间，特别是当外部组件的实现还未完成时。通常，大多数基本组件都不能准时可用。如果这些组件不在适当位置你就不能测试你自己的代码的话，那么你就在朝灾难走去。你的客户不会关心你只有两个小时来集成迟到了两周的组件。他们知道的全部就是整套产品被延期了和这是违约的。

原则 7：优先编写测试代码

优先编写测试代码是标准的 XP 方法，但却总会忽视它。假设你正在努力编写正确的代码，那么简单地从推迟编写测试代码中节省的时间只是一个幻想。注意，这不是说你应该一次性编写全部测试代码后，再一次性全部实现。编写一些测试代码，实现它们，再编写一些测试代码，再实现它们是个更好的办法，设计以这种方式得以进展。在实现阶段捕捉错误并在下一组测试中改正它。以这种方式编写测试也变得更容易一些。只需一点点努力，就可能容易地对任何程序进行彻底的测试。当然，不可避免存在这些原则不适用的情况，于是，看起来好像不可能对功能进行测试。当出现这些情况时，我们要尽力退一步地看这个问题，“我怎样才可能测试这种代码？”相反地，我们要问一下自己，“我怎样才能以可测试方式编写这些代码呢？”这种想法上的改变的结果经常是增加了大量仅仅服务于简化测试的功能。但是出现这种情况完全正常。就像很多现有的设计模式，它们只是为了增加程序的可扩展性，比如往程序中添加很多类，所以开发简化测试的新模式是可以接受的。实际上，面向对象语言的很多特征都是为了简化扩展而包含进去的。那么语言的未来版本（或全新的语言）是可以包含简化测试的特征的（比如 Java 语言）。人们计划在未来版本中包含很多更强大的类型系统、断言（assertion）等等，就像面向对象的语言已经增加了我们重用和扩展现有代码的程度一样，将来，面向测试的设计和特征将帮助我们增强新老代码的健壮性。

附录C　软件测试术语表

根据 ISTQB（国际软件测试资质认证委员会）提供的软件测试标准软件测试术语表翻译而成。本中文版不是 ISTQB 的官方的翻译版本，只是由一些软件测试的爱好者出于对软件测试的兴趣自发的翻译。我们无法保证中文版和英文版的一致性，同时也不保证提供的信息的正确性和完整性。如果有任何的建议或意见，请发邮件到：mailto:skyqa@skyqa.com。

A

- Abstract test case（High level test case）：概要测试用例
- Acceptance：验收
- Acceptance criteria：验收标准
- Acceptance testing：验收测试
- Accessibility testing：易用性测试
- Accuracy：精确性
- Actual outcome（actual result）：实际输出/实际结果
- Ad hoc review（informal review）：非正式评审
- Ad hoc testing：随机测试
- Adaptability：自适应性
- Agile testing：敏捷测试
- Algorithm test（branch testing）：分支测试
- Alpha testing：alpha 测试
- Analyzability：易分析性
- Analyzer：分析员
- Anomaly：异常
- Arc testing：分支测试
- Attractiveness：吸引力
- Audit：审计
- Audit trail：审计跟踪
- Automated testware：自动测试组件
- Availability：可用性

B

- Back-to-back testing：对比测试
- Baseline：基线
- Basic block：基本块
- Basis test set：基本测试集
- BeBugging：错误撒播
- Behavior：行为

- Benchmark test：基准测试
- Bespoke software：定制的软件
- Best practice：最佳实践
- Beta testing：Beta 测试
- Big-bang testing：集成测试
- Black-box technique：黑盒技术
- Black-box testing：黑盒测试
- Black-box test design technique：黑盒测试设计技术
- Blocked test case：被阻塞的测试用例
- Bottom-up testing：自底向上测试
- Boundary value：边界值
- Boundary value analysis：边界值分析
- Boundary value coverage：边界值覆盖率
- Boundary value testing：边界值测试
- Branch：分支
- Branch condition：分支条件
- Branch condition combination coverage：分支条件组合覆盖率
- Branch condition combination testing：分支条件组合测试
- Branch condition coverage：分支条件覆盖率
- Branch coverage：分支覆盖率
- Branch testing：分支测试
- Bug：缺陷
- Business process-based testing：基于商业流程的测试

C

- Capability Maturity Model（CMM）：能力成熟度模型
- Capability Maturity Model Integration（CMMI）：集成能力成熟度模型
- Capture/playback tool：捕获/回放 工具
- Capture/replay tool：捕获/重放 工具
- Computer Aided Software Engineering（CASE）：电脑辅助软件工程
- Computer Aided Software Testing（CAST）：电脑辅助软件测试
- Cause-effect graph：因果图
- Cause-effect graphing：因果图技术
- Cause-effect analysis：因果分析
- Cause-effect decision table：因果判定表
- Certification：认证
- Changeability：可变性
- Change control：变更控制
- Change control board：变更控制委员会
- Checker：检查人员
- Chow's coverage metrics（N-switch coverage）：N 切换覆盖率

- Classification tree method：分类树方法
- Code analyzer：代码分析器
- Code coverage：代码覆盖率
- Code-based testing：基于代码的测试
- Co-existence：共存性
- Commercial off-the-shelf software：商用离岸软件
- Comparator：比较器
- Compatibility testing：兼容性测试
- Compiler：编译器
- Complete testing：完全测试/穷尽测试
- Completion criteria：完成标准
- Complexity：复杂性
- Compliance：一致性
- Compliance testing：一致性测试
- Component：组件
- Component integration testing：组件集成测试
- Component specification：组件规格说明
- Component testing：组件测试
- Compound condition：组合条件
- Concrete test case（low level test case）：详细测试用例
- Concurrency testing：并发测试
- Condition：条件表达式
- Condition combination coverage：条件组合覆盖率
- Condition coverage：条件覆盖率
- Condition determination coverage：条件判定覆盖率
- Condition determination testing：条件判定测试
- Condition testing：条件测试
- Condition outcome：条件结果
- Confidence test（smoke test）：信心测试（冒烟测试）
- Configuration：配置
- Configuration auditing：配置审核
- Configuration control：配置控制
- Configuration control board（CCB）：配置控制委员会
- Configuration identification：配置标识
- Configuration item：配置项
- Configuration management：配置管理
- Configuration testing：配置测试
- Confirmation testing：确认测试
- Conformance testing：一致性测试
- Consistency：一致性

- Control flow：控制流
- Control flow graph：控制流图
- Control flow path：控制流路径
- Conversion testing：转换测试
- COTS（Commercial Off-The-Shelf software）：商业离岸软件
- Coverage：覆盖率
- Coverage analysis：覆盖率分析
- Coverage item：覆盖项
- Coverage tool：覆盖率工具
- Custom software：定制软件
- Cyclomatic complexity：圈复杂度
- Cyclomatic number：圈数

D

- Daily build：每日构建
- Data definition：数据定义
- Data driven testing：数据驱动测试
- Data flow：数据流
- Data flow analysis：数据流分析
- Data flow coverage：数据流覆盖率
- Data flow test：数据流测试
- Data integrity testing：数据完整性测试
- Database integrity testing：数据库完整性测试
- Dead code：无效代码
- DeBugger：调试器
- DeBugging：调试
- DeBugging tool：调试工具
- Decision：判定
- Decision condition coverage：判定条件覆盖率
- Decision condition testing：判定条件测试
- Decision coverage：判定覆盖率
- Decision table：判定表
- Decision table testing：判定表测试
- Decision testing：判定测试技术
- Decision outcome：判定结果
- Defect：缺陷
- Defect density：缺陷密度
- Defect Detection Percentage（DDP）：缺陷发现率
- Defect management：缺陷管理
- Defect management tool：缺陷管理工具
- Defect masking：缺陷屏蔽

- Defect report：缺陷报告
- Defect tracking tool：缺陷跟踪工具
- Definition-use pair：定义-使用对
- Deliverable：交付物
- Design-based testing：基于设计的测试
- Desk checking：桌面检查
- Development testing：开发测试
- Deviation：偏差
- Deviation report：偏差报告
- Dirty testing：负面测试
- Documentation testing：文档测试
- Domain：域
- Driver：驱动程序
- Dynamic analysis：动态分析
- Dynamic analysis tool：动态分析工具
- Dynamic comparison：动态比较
- Dynamic testing：动态测试

E

- Efficiency：效率
- Efficiency testing：效率测试
- Elementary comparison testing：基本组合测试
- Emulator：仿真器、仿真程序
- Entry criteria：入口标准
- Entry point：入口点
- Equivalence class：等价类
- Equivalence partition：等价区间
- Equivalence partition coverage：等价区间覆盖率
- Equivalence partitioning：等价划分技术
- Error：错误
- Error guessing：错误猜测技术
- Error seeding：错误撒播
- Error tolerance：错误容限
- Evaluation：评估
- Exception handling：异常处理
- Executable statement：可执行的语句
- Exercised：可执行的
- Exhaustive testing：穷尽测试
- Exit criteria：出口标准
- Exit point：出口点
- Expected outcome：预期结果

- Expected result：预期结果
- Exploratory testing：探测测试

F

- Fail：失败
- Failure：失败
- Failure mode：失败模式
- Failure Mode and Effect Analysis（FMEA）：失败模式和影响分析
- Failure rate：失败频率
- Fault：缺陷
- Fault density：缺陷密度
- Fault Detection Percentage（FDP）：缺陷发现率
- Fault masking：缺陷屏蔽
- Fault tolerance：缺陷容限
- Fault tree analysis：缺陷树分析
- Feature：特征
- Field testing：现场测试
- Finite state machine：有限状态机
- Finite state testing：有限状态测试
- Formal review：正式评审
- Frozen test basis：测试基线
- Function Point Analysis（FPA）：功能点分析
- Functional integration：功能集成
- Functional requirement：功能需求
- Functional test design technique：功能测试设计技术
- Functional testing：功能测试
- Functionality：功能性
- Functionality testing：功能性测试

G

- glass box testing：白盒测试

H

- Heuristic evaluation：启发式评估
- High level test case：概要测试用例
- Horizontal traceability：水平跟踪

I

- Impact analysis：影响分析
- Incremental development model：增量开发模型
- Incremental testing：增量测试
- Incident：事件
- Incident management：事件管理
- Incident management tool：事件管理工具

- Incident report：事件报告
- Independence：独立
- Infeasible path：不可行路径
- Informal review：非正式评审
- Input：输入
- Input domain：输入范围
- Input value：输入值
- Inspection：审查
- Inspection leader：审查组织者
- Inspector：审查人员
- Installability：可安装性
- Installability testing：可安装性测试
- Installation guide：安装指南
- Installation wizard：安装向导
- Instrumentation：插装
- Instrumenter：插装工具
- Intake test：入口测试
- Integration：集成
- Integration testing：集成测试
- Integration testing in the large：大范围集成测试
- Integration testing in the small：小范围集成测试
- Interface testing：接口测试
- Interoperability：互通性
- Interoperability testing：互通性测试
- Invalid testing：无效性测试
- Isolation testing：隔离测试
- Item transmittal report：版本发布报告
- Iterative development model：迭代开发模型

K

- Key performance indicator：关键绩效指标
- Keyword driven testing：关键字驱动测试

L

- Learnability：易学性
- Level test plan：等级测试计划
- Link testing：组件集成测试
- Load testing：负载测试
- Logic-coverage testing：逻辑覆盖测试
- Logic-driven testing：逻辑驱动测试
- Logical test case：逻辑测试用例
- Low level test case：详细测试用例

M

- Maintenance：维护
- Maintenance testing：维护测试
- Maintainability：可维护性
- Maintainability testing：可维护性测试
- Management review：管理评审
- Master test plan：综合测试计划
- Maturity：成熟度
- Measure：度量
- Measurement：度量
- Measurement scale：度量粒度
- Memory leak：内存泄露
- Metric：度量
- Migration testing：移植测试
- Milestone：里程碑
- Mistake：错误
- Moderator：仲裁员
- Modified condition decision coverage：改进的条件判定覆盖率
- Modified condition decision testing：改进的条件判定测试
- Modified multiple condition coverage：改进的多重条件判定覆盖率
- Modified multiple condition testing：改进的多重条件判定测试
- Module：模块
- Module testing：模块测试
- Monitor：监视器
- Multiple condition：多重条件
- Multiple condition coverage：多重条件覆盖率
- Multiple condition testing：多重条件测试
- Mutation analysis：变化分析
- Mutation testing：变化测试

N

- N-switch coverage：N 切换覆盖率
- N-switch testing：N 切换测试
- Negative testing：负面测试
- Non-conformity：不一致
- Non-functional requirement：非功能需求
- Non-functional testing：非功能测试
- Non-functional test design techniques：非功能测试设计技术

O

- Off-the-shelf software：离岸软件
- Operability：可操作性

- Operational environment：操作环境
- Operational profile testing：运行剖面测试
- Operational testing：操作测试
- Oracle：标准
- Outcome：输出/结果
- Output：输出
- Output domain：输出范围
- Output value：输出值

P

- Pair programming：结队编程
- Pair testing：结队测试
- Partition testing：分割测试
- Pass：通过
- Pass/fail criteria：通过/失败 标准
- Path：路径
- Path coverage：路径覆盖
- Path sensitizing：路径敏感性
- Path testing：路径测试
- Peer review：同行评审
- Performance：性能
- Performance indicator：绩效指标
- Performance testing：性能测试
- Performance testing tool：性能测试工具
- Phase test plan：阶段测试计划
- Portability：可移植性
- Portability testing：移植性测试
- Postcondition：结果条件
- Post-execution comparison：运行后比较
- Precondition：初始条件
- Predicted outcome：预期结果
- Pretest：预测试
- Priority：优先级
- Probe effect：检测成本
- Problem：问题
- Problem management：问题管理
- Problem report：问题报告
- Process：流程
- Process cycle test：处理周期测试
- Product risk：产品风险
- Project：项目

- Project risk：项目风险
- Program instrumenter：编程工具
- Program testing：程序测试
- Project test plan：项目测试计划
- Pseudo-random：伪随机

Q

- Quality：质量
- Quality assurance：质量保证
- Quality attribute：质量属性
- Quality characteristic：质量特征
- Quality management：质量管理

R

- Random testing：随机测试
- Recorder：记录员
- Record/playback tool：记录/回放 工具
- Recoverability：可复原性
- Recoverability testing：可复原性测试
- Recovery testing：可复原性测试
- Regression testing：回归测试
- Regulation testing：一致性测试
- Release note：版本说明
- Reliability：可靠性
- Reliability testing：可靠性测试
- Replaceability：可替换性
- Requirement：需求
- Requirements-based testing：基于需求的测试
- Requirements management tool：需求管理工具
- Requirements phase：需求阶段
- Resource utilization：资源利用
- Resource utilization testing：资源利用测试
- Result：结果
- Resumption criteria：继续测试标准
- Re-testing：再测试
- Review：评审
- Reviewer：评审人员
- Review tool：评审工具
- Risk：风险
- Risk analysis：风险分析
- Risk-based testing：基于风险的测试
- Risk control：风险控制

- Risk identification：风险识别
- Risk management：风险管理
- Risk mitigation：风险消减
- Robustness：健壮性
- Robustness testing：健壮性测试
- Root cause：根本原因

S

- Safety：安全
- Safety testing：安全性测试
- Sanity test：健全测试
- Scalability：可测量性
- Scalability testing：可测量性测试
- Scenario testing：情景测试
- Scribe：记录员
- Scripting language：脚本语言
- Security：安全性
- Security testing：安全性测试
- Serviceability testing：可维护性测试
- Severity：严重性
- Simulation：仿真
- Simulator：仿真程序、仿真器
- Site acceptance testing：定点验收测试
- Smoke test：冒烟测试
- Software：软件
- Software feature：软件功能
- Software quality：软件质量
- Software quality characteristic：软件质量特征
- Software test incident：软件测试事件
- Software test incident report：软件测试事件报告
- Software Usability Measurement Inventory（SUMI）：软件可用性调查问卷
- Source statement：源语句
- Specification：规格说明
- Specification-based testing：基于规格说明的测试
- Specification-based test design technique：基于规格说明的测试设计技术
- Specified input：特定输入
- Stability：稳定性
- Standard software：标准软件
- Standards testing：标准测试
- State diagram：状态图
- State table：状态表

- State transition：状态迁移
- State transition testing：状态迁移测试
- Statement：语句
- Statement coverage：语句覆盖
- Statement testing：语句测试
- Static analysis：静态分析
- Static analysis tool：静态分析工具
- Static analyzer：静态分析工具
- Static code analysis：静态代码分析
- Static code analyzer：静态代码分析工具
- Static testing：静态测试
- Statistical testing：统计测试
- Status accounting：状态统计
- Storage：资源利用
- Storage testing：资源利用测试
- Stress testing：压力测试
- Structure-based techniques：基于结构的技术
- Structural coverage：结构覆盖
- Structural test design technique：结构测试设计技术
- Structural testing：基于结构的测试
- Structured walkthrough：面向结构的走查
- Stub：桩
- Subpath：子路径
- Suitability：符合性
- Suspension criteria：暂停标准
- Syntax testing：语法测试
- System：系统
- System integration testing：系统集成测试
- System testing：系统测试

T

- Technical review：技术评审
- Test：测试
- Test approach：测试方法
- Test automation：测试自动化
- Test basis：测试基础
- Test bed：测试环境
- Test case：测试用例
- Test case design technique：测试用例设计技术
- Test case specification：测试用例规格说明
- Test case suite：测试用例套

- Test charter：测试宪章
- Test closure：测试结束
- Test comparator：测试比较工具
- Test comparison：测试比较
- Test completion criteria：测试比较标准
- Test condition：测试条件
- Test control：测试控制
- Test coverage：测试覆盖率
- Test cycle：测试周期
- Test data：测试数据
- Test data preparation tool：测试数据准备工具
- Test design：测试设计
- Test design specification：测试设计规格说明
- Test design technique：测试设计技术
- Test design tool：测试设计工具
- Test driver：测试驱动程序
- Test driven development：测试驱动开发
- Test environment：测试环境
- Test evaluation report：测试评估报告
- Test execution：测试执行
- Test execution automation：测试执行自动化
- Test execution phase：测试执行阶段
- Test execution schedule：测试执行进度表
- Test execution technique：测试执行技术
- Test execution tool：测试执行工具
- Test fail：测试失败
- Test generator：测试生成工具
- Test leader：测试负责人
- Test harness：测试组件
- Test incident：测试事件
- Test incident report：测试事件报告
- Test infrastructure：测试基础组织
- Test input：测试输入
- Test item：测试项
- Test item transmittal report：测试项移交报告
- Test level：测试等级
- Test log：测试日志
- Test logging：测试记录
- Test manager：测试经理
- Test management：测试管理

- Test management tool：测试管理工具
- Test Maturity Model（TMM）：测试成熟度模型
- Test monitoring：测试跟踪
- Test object：测试对象
- Test objective：测试目的
- Test oracle：测试标准
- Test outcome：测试结果
- Test pass：测试通过
- Test performance indicator：测试绩效指标
- Test phase：测试阶段
- Test plan：测试计划
- Test planning：测试计划
- Test policy：测试方针
- Test Point Analysis（TPA）：测试点分析
- Test procedure：测试过程
- Test procedure specification：测试过程规格说明
- Test process：测试流程
- Test Process Improvement（TPI）：测试流程改进
- Test record：测试记录
- Test recording：测试记录
- Test reproduceability：测试可重现性
- Test report：测试报告
- Test requirement：测试需求
- Test run：测试运行
- Test run log：测试运行日志
- Test result：测试结果
- Test scenario：测试场景
- Test script：测试脚本
- Test set：测试集
- Test situation：测试条件
- Test specification：测试规格说明
- Test specification technique：测试规格说明技术
- Test stage：测试阶段
- Test strategy：测试策略
- Test suite：测试套
- Test summary report：测试总结报告
- Test target：测试目标
- Test tool：测试工具
- Test type：测试类型
- Testability：可测试性

● Testability review：可测试性评审
● Testable requirements：需求可测试性
● Tester：测试人员
● Testing：测试
● Testware：测试组件
● Thread testing：组件集成测试
● Time behavior：性能
● Top-down testing：自顶向下的测试
● Traceability：可跟踪性

U

● Understandability：易懂性
● Unit：单元
● unit testing：单元测试
● Unreachable code：执行不到的代码
● Usability：易用性
● Usability testing：易用性测试
● Use case：用户用例
● Use case testing：用户用例测试
● User acceptance testing：用户验收测试
● User scenario testing：用户场景测试
● User test：用户测试

V

● V-model：V 模式
● Validation：确认
● Variable：变量
● Verification：验证
● Vertical traceability：垂直可跟踪性
● Version control：版本控制
● Volume testing：容量测试

W

● Walkthrough：走查
● White-box test design technique：白盒测试设计技术
● White-box testing：白盒测试
● Wide Band Delphi：Delphi 估计方法

附录D 参 考 文 献

[1] 朱少民，软件测试方法和技术，清华大学出版社，2005.7

[2] 朱少民，合程软件测试，电子工业出版社，2007.9

[3] 赵斌，软件测试技术经典教程，科学出版社，2007.5

[4] 杨海燕 等译，软件度量，机械工业出版社，2004.9

[5] 陈能技译，软件测试技术大全，人民邮电出版社，2008.6

[6] 于涌译，软件性能测试与 LoadRunner 实战，人民邮电出版社，2008.5

[7] （美）William E.Perry，Effective Methods for Software Testing,Second Edition，John Wiley & sons,Inc.，2004.3

[8] 张银奎，软件调试，电子工业出版社，2008.6

[9] 王海鹏译，快速测试，人民邮电出版社，2004.1

[10] http://www.51testing.com

[11] http://www.testage.net

[12] http://www.blog.csdn.net/KerryZhu

[13] http://www.studytesting.com/content/section/3_49.html